EPJ Web of Conferences **318**, 04012 (2025) https://doi.org/10.1051/epjconf/202531804012
AAPM-III 2025

Construction of systems of kinematics and dynamics equations in mathematical rigging models for immersive environments

Amelia Grace [1,2,4*], *Igor* Kovalev[2,3,4,5], *Dmitry* Kovalev[2,6], *Anna* Voroshilova[1,4], and *Dmitry* Borovinsky[7]

[1]Krasnoyarsk Science and Technology City Hall of the Russian Union of Scientific and Engineering Public Associations, Krasnoyarsk, Russia
[2]Krasnoyarsk State Agrarian University, Krasnoyarsk, Russia
[3]Navoi State University of Mining and Technologies, Navoi, Uzbekistan
[4]Siberian Federal University, Krasnoyarsk, Russia
[5]Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, Russia
[6]National Research University "Tashkent Institute of Irrigation and Agricultural Mechanization Engineers", Tashkent, Uzbekistan
[7]FSBEE HE Siberian Fire and Rescue Academy EMERCOM of Russia, Zheleznogorsk, Russia

Abstract. The article discusses key aspects of constructing rigging equation systems in immersive environments based on physical laws, including kinematics and dynamics equations. It examines the mathematical models and methods underlying these systems, as well as their integration to create realistic and efficient virtual worlds. Special attention is given to the development of equations for mathematical rigging models that determine the operator's movement in a virtual nuclear power plant simulator. Taking into account the laws of physics in mathematical rigging models based on determining the structure of the user's skeleton and objects, as well as constructing kinematics and dynamics equations, allows characters to move naturally and react to interaction with the environment. The article is of interest to virtual reality specialists, and researchers involved in computing, mathematical modeling novel physics in virtual reality labs and computer graphics.

1 Introduction

Immersive systems are a complex set of software and hardware tools designed to simulate or model real objects, processes, or scenes in a virtual space using digital technologies. One of the applied areas of immersive systems is virtual reality (VR) – a technology that immerses the user in a simulated space by affecting their receptors (vision, hearing, smell, tactile sensations). Users can interact with this environment using devices equipped with motion sensors, creating a sense of presence that allows them to move and interact with objects in the virtual space [1].

---

* Corresponding author: alena.yabl@yandex.ru

The scenes and objects with which users interact in an immersive environment are based on positional tracking technology. Motion tracking technologies determine the position and orientation of a real object in the virtual environment, enabling the VR system to adapt to and respond to the user's movements, providing a more realistic and immersive experience. The determination of the position and orientation of a real object in space is achieved using special sensors and markers. Sensors capture signals from the real object as it moves and transmit the obtained information to the computer. Virtual reality glasses, equipped with gyroscopes, accelerometers, and infrared sensors, track the user's head movements. Additionally, external cameras and infrared beacons are used. However, to ensure that virtual systems provide a realistic experience, aspects such as correct formalization and construction of rigging equation systems must be considered to handle interactivity mechanics, user safety, coordination of multi-user scenarios, spatial awareness, and performance optimization [2, 3].

2 Materials and methods

Positional Tracking is a technology used in virtual reality to determine the position and orientation of the user in space. Positional tracking systems utilize sensors, cameras, or laser markers to track the position and movement of the headset in real-time, enabling the virtual environment to respond to the user's movements and providing appropriate changes in perspective and object display in the virtual space [4].

Rigging is the process of creating a skeletal structure or bone system (rigs) and attaching them to 3D models, which allows the models to be animated, controlled in their movements, and deformed. The skeletal structure consists of bones connected by joints, and mesh deformations are applied to the objects so they can be animated according to the movements of the bones. Rigging is a crucial stage as it allows for controlling and modeling the movement of objects in space. Rigging involves constructing a system of equations that describe how parts of the model will move in response to user actions or pre-set animations [5, 6].

Mechanics refer to the ways the user interacts with the virtual environment and the objects within it. They determine how the user can move, interact with, and influence objects in the virtual world.

Positional tracking, rigging, and mechanics are interconnected. Positional tracking provides real-time data on the position and orientation of the user and objects, which are used in rigging systems to animate virtual models, ensuring accurate and realistic movement. Rigging defines how models will move and deform. Mechanics use these movements and add physical laws and interaction rules to ensure the correct and plausible behavior of the models. Tracking data provides up-to-date information on the position and movement of the user and objects. Mechanics use this data to implement interactive scenarios and physical simulations, allowing the system to respond to user actions and changes in the virtual environment in real-time [7, 8].

Building rigging equation systems in virtual simulators requires a complex approach. One of the key stages is the development of mathematical models for scenes and objects. This stage involves creating and optimizing equations and algorithms that will be used to control the movements of users and objects and track their positions.

The development of mathematical models consists of two steps – creating rigging models and positional tracking models. The creation of rigging models is further subdivided into several stages – determining the skeleton structure, composing kinematic and dynamic equations.

At the stage of creating the skeleton structure, the anatomy of real objects is studied to identify key links and joints, on which the hierarchical skeleton structure is built. Each bone (link) is connected to a parent bone, creating a tree-like structure. Connections between bones, including joints and pivot points, are determined. Based on the collected data, bones are placed inside the 3D model of the object to ensure correct deformation during animation. The initial orientation of the bones, corresponding to the object's starting position, is set. For mesh deformation, weight maps are determined to correctly distribute the influence of the bones on the object's shape during movement. Skinning modifiers are applied for smooth transitions between deformations caused by different bones.

Kinematic equations are a key element in the rigging system as they allow for calculating the position and orientation of the model's parts based on joint rotation angles. The equations are divided into two main types: forward kinematics and inverse kinematics. Forward kinematics is used to calculate the position and orientation of the model's end links based on known joint angles and link lengths. This approach is used in most immersive systems. Kinematic equations include transformation matrices and cumulative transformation matrices. For each joint and bone, a transformation matrix, including rotations and translations, is calculated. The cumulative transformation, responsible for the position and orientation of the endpoint, is determined by sequentially applying the transformation matrices from the root joint to the end link.

Dynamic equations are used to calculate the forces and moments acting on each bone and joint. These equations model the movement and interaction between elements of virtual objects or characters, allowing for the creation of realistic animations and simulations in virtual environments. By applying dynamic equations, the behavior of objects can be analyzed and predicted based on external forces applied to them and their internal characteristics.

3 Results and discussion

Virtual simulators of nuclear power plants are critically important for several reasons. First, they ensure safety by allowing users to undergo training and practice actions in a safe environment, without putting themselves or others at risk. VR simulators enable the modeling of various emergency situations and train operators in the correct actions to prevent and mitigate accidents, thereby minimizing the likelihood of human errors in real conditions. Second, virtual simulators contribute to cost savings. Training on virtual simulators is significantly cheaper than conducting training on actual facilities, reducing the need to use real equipment for training, which decreases wear and tear and the associated maintenance and repair costs. Third, the relevance of this IT strategy lies in enhancing training effectiveness. Operators can repeatedly practice various scenarios, which helps them better absorb and consolidate the knowledge and skills acquired. Modern simulators can accurately reproduce real working conditions, including the physics of processes and control interfaces, making the training more effective [9-12].

Fourth, there is the possibility of control and assessment. Instructors can monitor and evaluate the actions of operators in real time, providing feedback and correcting mistakes. Simulators allow precise recording of operators' actions and assessment based on objective criteria, contributing to a fairer and more accurate evaluation. Adaptation to new technologies is also an important element of integration. Virtual simulators can be quickly updated and adapted to new technological solutions and changes in nuclear power plant equipment. Operators can familiarize themselves in advance with new systems and technologies that will be implemented at real plants, accelerating the process of modernization and technology deployment. Working at a nuclear power plant is associated with high responsibility and stress. Simulators help operators get used to stressful conditions and develop the skills to make correct decisions under pressure.

Ultimately, virtual simulators of nuclear power plants play a key role in ensuring the safe, efficient, and economically viable operation of nuclear power plants, contributing to the improvement of operators' qualifications and the reduction of accident risks.

In a virtual simulator of a nuclear power plant, the user interacts with numerous important objects and systems that simulate real equipment and working conditions at the plant. One of the main elements of interaction is the control panels, where the operator can manage various operating parameters, start and stop processes, and monitor the operation of the reactor, turbines, and other key systems. Buttons, switches, and levers allow the regulation of various mechanisms and systems, from electrical circuits to cooling and safety systems.

Screens and monitors display important data, such as temperature, pressure, radiation levels, and the status of various systems, providing the operator with the necessary information to make decisions and respond to changes in the plant's operation. Tools and equipment, including wrenches, screwdrivers, and measuring instruments, are presented in the virtual environment for performing maintenance and repairs. Virtual doors and hatches provide access to various areas of the plant, such as the reactor hall or machine rooms, allowing the user to explore and manage different aspects of the plant's operation. Pipelines and valves that regulate the flows of water, steam, and other working fluids can be opened or closed by the operator depending on the required processes and conditions [13].

Emergency systems allow the operator to simulate responses to emergency situations and practice effective management of emergency procedures. Documentation and diagrams available electronically help the operator access reference information and familiarize themselves with operational and maintenance procedures. In some simulators, virtual colleagues may be present, with whom the user can interact, solving tasks collaboratively and coordinating actions. All these elements simulate realistic working conditions at a nuclear power plant, allowing the operator to practice skills, improve responses to various situations, and enhance professional training levels without the need for real equipment and without putting themselves and others at risk [14].

Creation of rigging models for the operator of a virtual nuclear power plant simulator.

3.1 Skeletal structure

3.1.1 The anatomy of the operator

The anatomy of the operator consists of the following elements:

- Head and Neck: the head serves as the center for perception and control coordination, being flexibly connected to the neck. This connection allows for head turns and tilts to view the surroundings and interact with control system interfaces;
- Torso and Spine: the torso consists of the spine, which provides fundamental support for the limbs and internal organs. the spine enables bending, twisting, and other movements to ensure the operator's mobility;
- Limbs (Arms and Legs): the operator's arms (shoulders, forearms, and hands) are essential for performing tasks. The shoulder joints connect the arms to the torso, providing support and mobility. Elbows and wrists allow the operator to perform complex movements and manipulations with interfaces and equipment. The legs (thighs, shins, and feet) are connected to the torso through the hip joints, which provide support and the ability to move within the virtual environment. Knee and ankle joints enable movement and positional changes of the legs for various operations;

- Joints: joints allow different parts of the body to connect and perform movements. Each joint has its own characteristics and limitations, which determine the range of motion and angles of rotation;
- Operator Movements: the operator's movements include head turns to monitor processes at the station, raising and moving limbs to interact with control panels and equipment, and moving around the workspace.

3.1.2 Hierarchy of bones

Neck – Head: see how the head rotates and tilts in relation to the neck.

Torso – Neck: allows the neck to rotate and tilt relative to the torso.

Torso – Shoulders: connects the shoulders to the torso, giving mobility to the arms and their position relative to the torso.

Shoulders – Forearms: look at bending and extending the elbow joints.

Forearms – Hands: determines the position and orientation of the hands, including wrist flexion and extension.

Torso – Hip joints: provides the ability to telephone and lift the legs.

Hip joints – Thighs: connects the hips to the hip joints, causing the legs to move forward and backward.

Hips – Shins: look at the flexion and extension of the knee joints.

Shin – Feet: determines the position and orientation of the foot, including flexion and extension of the ankle joints.

3.1.3 Pivot points and movement restrictions

Head: rotation point – upper cervical vertebra, movement restrictions – ability to turn left and right, tilt up and down. Limits can be set to prevent excessive rotation and tilt angles.

Neck: rotation point – middle cervical vertebra, motion constraints – rotation and tilt constraints to maintain natural head animation.

Shoulders: rotation point – shoulder joint, movement restrictions: up, down, forward and backward movement; restrictions to prevent overextension or underextension of the shoulder.

Elbow joints: rotation point: elbow joint; movement restrictions: flexion and extension of the arm. Limitations may include maximum flexion and extension angles.

Hip joints: rotation point – hip joint, movement restrictions – rotation of the leg at the hip joint, lifting the leg forward, backward and to the sides. Constraints are set to simulate the natural movements of the human body.

Knee joints: rotation point – knee joint; movement restrictions – flexion and extension of the leg, limitations may include maximum knee flexion and extension angles.

Ankle joints: rotation point – ankle joint, movement restrictions – flexion and extension of the foot, rotation around an axis. Constraints are set to ensure stability and realistic foot movements.

3.1.4 Mesh deformation

In a virtual simulator of a nuclear power plant, the operator is represented as a 3D model, which consists of a polygonal mesh (mesh). To animate this model, weight maps are used to determine how each bone in the operator's skeleton will contribute to the deformation of the polygon mesh.

Taking into account the average height and weight of a male operator, the average values of the weight cards are:

- Operator's head. For an operator with a height of 165 to 180 cm at the top of the mesh, the corresponding head regions can have a weight between 0.7 and 1.0. Vertices around the top of the head: 0.8, vertices around the neck: 0.7, vertices around the face: 0.6. These values will allow the head to adequately respond to movements, maintaining its shape and natural rotation.
- Operator's torso. The mesh vertices that make up the operator's body can have a weight from 0.4 to 0.8. Peaks on the back: 0.6, peaks on the sides: 0.5, peaks on the belly: 0.4. The values will ensure natural bending of the torso when bending and turning.
- Operator's hands. Mesh vertices on the operator's arm can have a weight from 0.5 to 0.7. Shoulder vertices: 0.7, biceps vertices: 0.6, forearm vertices: 0.5. The specified values will ensure natural hand movement.
- Operator's legs. The mesh vertices on the operator's leg can have a weight from 0.5 to 0.7. Tops of thigh: 0.7, tops of shins: 0.6, tops of foot: 0.5. The values will ensure the correct deformation of the leg when walking or bending the knee joint.
- Operator suit. The vertices of the mesh, which are located in the area of the operator's suit, can have a uniform weight, for example, from 0.3 to 0.5, depending on how tightly the suit fits the operator's body, which will maintain the shape of the suit during movements and prevent unwanted deformations.

The given weight map values distribute the influence of each bone across the vertices of the mesh to ensure natural and realistic operator movement in the virtual environment.

3.1.5 Skin modifiers

When using skin modifiers, weight maps are applied to smoothly transition between deformities caused by different bones. For example, if the operator turns his head to the left, then the LBS modifier uses high weights on the vertices of the head mesh to properly animate the head rotation. The operator raises his arm, then the weight maps for the arm and shoulder bones are activated so that the vertices of the arm and shoulder mesh move to the appropriate position without distortion. The operator tilts the torso forward: the weight values for the torso bones allow the torso mesh to smoothly change its shape and position.

3.2 Equations of direct kinematics

Forward kinematics equations are used to calculate the position and orientation of the model's end links based on known joint angles and link lengths.

3.2.1 Transformation matrices

For each joint and bone, a transformation matrix is calculated, which includes rotations and displacements. The transformation matrix for each joint or bone in a hierarchical structure (such as an operator's skeleton) describes the transformations that define the position and orientation of that bone relative to its parent bone (rotation, translation, and, if necessary, scaling).

Each bone i in the skeletal hierarchy has its own transformation matrix T_i, which determines its position and orientation relative to its parent bone. The bone transformation matrices are combined sequentially to determine the final position and orientation of the final bone in the hierarchy. For each joint and bone of the operator, a transformation matrix is calculated, which includes rotations and displacements.

Transformation matrix T_i for the joint i is defined as the product of the displacement and rotation matrices, including rotation by angle θ around the axes x, y, z and displacement by vectors *dx, dy, dz.*

Transformation matrix T_i expressed as the product of the rotation and displacement matrices $T_i = T_{translate}(dx_i, dy_i, dz_i) \cdot R_z(\theta_{i,z}) \cdot R_y(\theta_{i,y}) \cdot R_x(\theta_{i,x})$, where:

- $T_{translate}(dx_i, dy_i, dz_i)$ – translation matrix describing the displacement of bone i relative to its parent bone;
- $R_z(\theta_{i,z}), R_y(\theta_{i,y}), R_x(\theta_{i,x})$ – rotation matrix around the *x, y, z* axes, respectively.

For each pair of operator bones, transformation matrices are created that include rotations and displacements, where:

- $h_{neck}, h_{torso}, h_{shoulder}, h_{hip}$– height of the corresponding parts of the body;
- $w_{shoulder}$– shoulder width;
- $l_{upper\_arm}, l_{forearm}, l_{hip}, l_{thigh}, l_{shin}$– lengths of body parts;
- $\theta_{head_x}, \theta_{head_y}, \theta_{head_z}$– angles of head rotation around axes *x, y, z*;
- $\theta_{neck_x}, \theta_{neck_y}, \theta_{neck_z}$– neck rotation angles around axes *x, y, z*;
- $\theta_{shoulder_x}, \theta_{shoulder_y}, \theta_{shoulder_z}$– angles of rotation of the arm around the axes *x, y, z*;
- $\theta_{elbow_x}, \theta_{elbow_y}, \theta_{elbow_z}$– angles of rotation of the elbow around the axes *x, y, z*;
- $\theta_{wrist_x}, \theta_{wrist_y}, \theta_{wrist_z}$– angles of rotation of the wrist around the axes *x, y, z*;
- $\theta_{hip_x}, \theta_{hip_y}, \theta_{hip_z}$– angles of rotation of the hip joint around the axes *x, y, z*;
- $\theta_{thigh_x}, \theta_{thigh_y}, \theta_{thigh_z}$– angles of rotation of the femur around its axes *x, y, z*;
- $\theta_{knee_x}, \theta_{knee_y}, \theta_{knee_z}$– angles of rotation of the knee around the axes *x, y, z*;
- $\theta_{ankle_x}, \theta_{ankle_y}, \theta_{ankle_z}$– angles of rotation of the ankle around the axes *x, y, z.*

Transformation matrices for couples:

- Neck – Head

$$T_{neck\_head} = T_{translate}(0, h_{neck}, 0) \cdot R_z(\theta_{head\_z}) \cdot R_y(\theta_{head\_y}) \cdot R_x(\theta_{head\_x})$$

- Torso – Neck

$$T_{torso\_neck} = Ttranslate(0, h_{torso}, 0) \cdot R_z(\theta_{neck\_z}) \cdot R_y(\theta_{neck\_y}) \cdot R_x(\theta_{neck\_x})$$

- Torso – Shoulders

$$T_{torso\_shoulders} = T_{translate}(0, h_{shoulder}, w_{shoulder}/2) \cdot R_z(\theta_{shoulder\_z}) \cdot R_y(\theta_{shoulder\_y}) \cdot R_x(\theta_{shoulder\_x})$$

- Shoulders – Forearms

$$T_{shoulder\_forearm} = T_{translate}(0,0, l_{upper\_arm}) \cdot R_z(\theta_{elbow\_z}) \cdot R_y(\theta_{elbow\_y}) \cdot R_x(\theta_{elbow\_x})$$

- Forearms – Hands

$$T_{forearm\_hand} = T_{translate}(0,0, l_{f_{orearm}}) \cdot R_z(\theta_{wrist\_z}) \cdot R_y(\theta_{wrist\_y}) \cdot R_x(\theta_{wrist\_x})$$

- Torso – Hip joints

$$T_{torso\_hip} = T_{translate}(0, h_{hip}, 0) \cdot R_z(\theta_{hip\_z}) \cdot R_y(\theta_{hip\_y}) \cdot R_x(\theta_{hip\_x})$$

- Hip joints – Thighs

$$T_{hip\_thigh} = T_{translate}(0,0, l_{hip}) \cdot R_z(\theta_{thigh\_z}) \cdot R_y(\theta_{thigh\_y}) \cdot R_x(\theta_{thigh\_x})$$

- Hips – Shins

$$T_{thigh\_shin} = T_{translate}(0,0, l_{thigh}) \cdot R_z(\theta_{knee\_z}) \cdot Ry(\theta_{knee\_y}) \cdot R_x(\theta_{knee\_x})$$

- Shins – Feet

$$T_{shin\_foot} = T_{translate}(0,0,l_{shin}) \cdot R_z(\theta_{ankle_z}) \cdot R_y\left(\theta_{ankle_y}\right) \cdot R_x(\theta_{ankle_x}).$$

3.2.2 Cumulative transformation

The position and orientation of the end point is determined by sequential application of transformation matrices from the root joint to the end link. Cumulative transformation T_{cum} for an end point will be the product of all transformation matrices on the path from the root to this point $T_{cum} = T_{root} \cdot T_{link1} \cdot T_{link2} \cdot \dots \cdot T_{linkn}$, where:

- T_{cum}– cumulative transformation matrix for the final link;
- T_{root}– transformation matrix for the root joint;
- T_{link_i}– transformation matrix for the *i*-th link in the chain.

Each transformation matrix T_{link_i} decomposes into the product of the rotation and displacement matrices $T_{link_i} = T_{translate}(x_i, y_i, z_i) \cdot R_z(\theta_{i_z}) \cdot R_y(\theta_{i_y}) \cdot R_x(\theta_{i_x})$, where

$T_{translate}(x_i, y_i, z_i)$ – translation matrix for the link *i* with coordinates (x_i, y_i, z_i);
$R_x(\theta_{i_x})$– rotation matrix around the *x* axis for link *i* with rotation angle θ_{i_x};
$R_y(\theta_{i_y})$ – rotation matrix around the *y* axis for link *i* with rotation angle θ_{i_y};
$R_z(\theta_{i_z})$ – rotation matrix around the *z* axis for link *i* with rotation angle θ_{i_z}.

Final equation for a chain of n links: $T_{cum} = (\prod_{i=1}^{n} \quad T_{link_i})$, where is each T_{link_i} includes broadcast and rotation for the corresponding link $T_{link_i} = T_{translate}(x_i, y_i, z_i) \cdot R_z(\theta_{i_z}) \cdot R_y(\theta_{iy}) \cdot R_x(\theta_{i_x})$.

Cumulative transformation equations for the chain of bones Torso – Neck – Head.

Transformation matrix from torso to neck:

$$T_{torso\_neck} = T_{translate}(0, h_{torso}, 0) \cdot R_z(\theta_{neck_z}) \cdot R_y(\theta_{neck_y}) \cdot R_x(\theta_{neck_x}) \,. \tag{1}$$

Transformation matrix from neck to head:

$$T_{neck\_head} = T_{translate}(0, h_{neck}, 0) \cdot R_z(\theta_{head_z}) \cdot R_y\left(\theta_{head_y}\right) \cdot R_x(\theta_{head_x}). \tag{2}$$

Cumulative transformation matrix from torso to head:

$$T_{torso\_head} = T_{torso\_neck} \cdot T_{neck\_head} \,. \tag{3}$$

Cumulative transformation equations for the chain of bones Torso – Shoulders – Forearms – Hands.

Transformation matrix from torso to shoulders:

$$T_{torso\_shoulders} = T_{translate}(0, h_{shoulder}, w_{shoulder}/2) \cdot R_z(\theta_{shoulder_z}) \cdot R_y(\theta_{shoulder_y}) \cdot R_x(\theta_{shoulder_x}).$$

Transformation matrix from shoulders to forearms:

$$T_{shoulder\_forearm} = T_{translate}(0,0, l_{upper\_arm}) \cdot R_z(\theta_{elbow_z}) \cdot R_y(\theta_{elbow_y}) \cdot R_x(\theta_{elbow_x})$$

Transformation matrix from forearms to hands:

$$T_{forearm\_hand} = T_{translate}(0,0, l_{forearm}) \cdot R_z(\theta_{wrist_z}) \cdot R_y(\theta_{wrist_y}) \cdot R_x(\theta_{wrist_x}).$$

Cumulative transformation matrix from torso to hands.

$$T_{torso\_hand} = T_{torso\_shoulders} \cdot T_{shoulder\_forearm} \cdot T_{forearm\_hand}.$$

Cumulative transformation equations for the chain of bones Torso – Hip joints – Hips – Lower legs – Feet.

Transformation matrix from torso to hip joints:

$$T_{torso\_hip} = T_{translate}(0, h_{hip}, 0) \cdot R_z(\theta_{hip_z}) \cdot R_y(\theta_{hip_y}) \cdot R_x(\theta_{hip_x}).$$

Transformation matrix from hips to thighs:

$$T_{hip\_thigh} = T_{translate}(0,0, l_{hip}) \cdot R_z(\theta_{thigh_z}) \cdot R_y(\theta_{thigh_y}) \cdot R_x(\theta_{thigh_x}).$$

Transformation matrix from hips to shins:

$$T_{thigh\_shin} = T_{translate}(0{,}0, l_{thigh}) \cdot R\ _z(\theta_{knee_z}) \cdot R_y\left(\theta_{knee_y}\right) \cdot R_x(\theta_{knee_x}).$$

Transformation matrix from hips to shins:

$$T_{shin\_foot} = T_{translate}(0,0,l_{\ shin}) \cdot R_z(\theta_{ankle_z}) \cdot R_y(\theta_{ankle_y}) \cdot R_x(\theta_{ankle_x})\ .$$

Cumulative transformation matrix from torso to feet:

$$T_{torso\_foot} = T_{torso\_hip} \cdot T_{hip\_thigh} \cdot T_{thigh\_shin} \cdot T_{shin\_foot}.$$

The final equation for the cumulative transformation matrix, starting from the torso to the final link of each chain (head, hands, feet), is as follows.

Cumulative transformation matrix from torso to head

$$\begin{aligned} T_{torso\_head} = &\left(T_{translate}(0, h_{torso}, 0) \cdot R_z(\theta_{neck_z}) \cdot R_y\left(\theta_{neck_y}\right) \cdot R_x(\theta_{neck_x})\right) \\ &\cdot \left(T_{translate}(0, h_{neck}, 0) \cdot R_z(\theta_{head_z}) \cdot R_y\left(\theta_{head_y}\right) \cdot R_x(\theta_{head_x})\right). \end{aligned}$$

Cumulative transformation matrix from the torso to the hands.

Left hand:

$$\begin{aligned} T_{torso\_right\_hand} = &(T_{translate}(0, h_{shoulder}, -\frac{w_{shoulder}}{2}) \cdot R_z(\theta_{shoulder_z}) \cdot R_y(\theta_{shoulder_y}) \\ &\cdot R_x(\theta_{shoulder_x})) \cdot \\ &(T_{translate}(0{,}0, l_{upper\backslash arm}) \cdot R_z(\theta_{elbow_z}) \cdot R_y(\theta_{elbow_y}) \cdot R_x(\theta_{elbow_x})) \\ &\cdot (T_{translate}(0{,}0, l_{forearm}) \cdot \\ &R_z(\theta_{wrist_z}) \cdot R_y(\theta_{wrist_y}) \cdot R_x(\theta_{wrist_x})). \end{aligned}$$

Right hand:

$$\begin{aligned} T_{torso\_right\_hand} = &(T_{translate}(0, h_{shoulder}, \frac{w_{shoulder}}{2}) \cdot R_z(\theta_{shoulder_z}) \cdot R_y(\theta_{shoulder_y}) \\ &\cdot R_x(\theta_{shoulder_x})) \cdot \\ &(T_{translate}(0{,}0, l_{upper\backslash arm}) \cdot R_z(\theta_{elbow_z}) \cdot R_y(\theta_{elbow_y}) \cdot R_x(\theta_{elbow_x})) \cdot \\ &(T_{translate}(0{,}0, l_{forearm}) \cdot R_z(\theta_{wrist_z}) \cdot R_y(\theta_{wrist_y}) \cdot R_x(\theta_{wrist_x})). \end{aligned}$$

Cumulative transformation matrix from torso to feet.

Left foot:

$$\begin{aligned} T_{torso\_right\_foot} = &(T_{translate}(0, h_{hip}, -\frac{w_{hip}}{2}) \cdot R_z(\theta_{hip_z}) \cdot R_y(\theta_{hip_y}) \cdot R_x(\theta_{hip_x})) \cdot (T_{translate}(0,0,l_{thigh}) \cdot R_z(\theta_{thigh_z}) \cdot R_y(\theta_{thigh_y}) \cdot \\ &R_x(\theta_{thighx})) \cdot (T_{translate}(0,0,l_{shin}) \cdot R_z(\theta_{\ knee_z}) \cdot R_y(\theta_{knee_y}) \cdot R_x(\theta_{kneex})) \cdot (T_{translate}(0,0,l_{\ foot}) \cdot R_z(\theta_{ankle_z}) \cdot R_y(\theta_{\ ankle_y}) \cdot R_x(\theta_{ankle_x})) \end{aligned}$$

Right foot:

$$\begin{aligned} T_{torso\_right\_foot} = &(T_{translate}(0, h_{hip}, \frac{w_{hip}}{2}) \cdot R_z(\theta_{hip_z}) \cdot R_y(\theta_{hip_y}) \cdot R_x(\theta_{hip_x})) \cdot (T_{translate}(0,0,l_{thigh}) \cdot R_z(\theta_{thigh_z}) \cdot R_y(\theta_{thigh_y}) \cdot \\ &R_x(\theta_{thighx})) \cdot (T_{translate}(0,0,l_{shin}) \cdot R_z(\theta_{\ knee_z}) \cdot R_y(\theta_{knee_y}) \cdot R_x(\theta_{kneex})) \cdot (T_{translate}(0,0,l_{\ foot}) \cdot R_z(\theta_{ankle_z}) \cdot R_y(\theta_{\ ankle_y}) \cdot R_x(\theta_{ankle_x})) \end{aligned}$$

General cumulative transformation matrix from the body to the end links

$$T_{total} = \{T_{torso\_head}, T_{torso\_right\_hand}, T_{torso\_left\_hand}, T_{torso\_right\_foot}, T_{torso\_left\_foot}\}.$$

These equations describe the cumulative transformation from the torso to each end point in 3D space, taking into account all intermediate links and their rotations.

3.3 Equations of dynamics

Dynamic equations are used to calculate the forces and moments acting on each bone and joint, including the use of equations of motion that account for mass, moments of inertia, forces, and moments acting on each part of the system. Let's consider the basic equations for a kinematic chain consisting of segments (bones) and joints (rotation points).

Newton's law of motion for translational motion $\sum F = ma$, where

- $\sum F$– the sum of external forces acting on an object;

- m– object mass;
- a– acceleration of the object's center of mass.

Newton's law of motion for rotational motion (moment of force)$\sum M = I\alpha$, where

- $\sum M$– the sum of external moments of forces acting on an object;
- I– moment of inertia of an object about the axis of rotation;
- α– angular acceleration.

Forward movement of each segment: for each segment$i$ $\sum F_i = m_i a_i$, where $\boldsymbol{a}_i$ – acceleration of the segment's center of mass *i*.

Rotational movement of each segment: for each segment i: $\sum M_i = I_i \alpha_i$, where I_i– segment moment of inertia i relative to its center of mass, α_i– segment angular acceleration.

Forces and moments at joints: $F_{i-1,i}$and $M_{i-1,i}$– force and moment acting on segment from the side of the segment$i-1$.

For each segment i:

- $\sum F_i = F_{i-1,i} - F_{i,i+1} + m_i g$, where $F_{i,i+1}$– force acting on segment$i$from the side of the segment, where *g* – gravitational acceleration.
- $\sum M_i = M_{i-1,i} - M_{i,i+1} + r_i \times F_{i-1,i}$, where $\boldsymbol{r}_i$ – vector connecting the center of mass of segment i with the point of force application $F_{i-1,i.}$.

For each chain of segments:

Translational equations for a segment i:$F_{i-1,i} - F_{\,i,i+1} + m_i g = m_i a_i$.

Rotational equations for a segment i: $M_{i-1,i} - M_{i,i+1} + r_i \times F_{i-1,i} = I_i \alpha_i$.

These equations apply to every part of the body, from the end links to the torso.

3.3.1 Dynamic equations for each of the chains

Applying dynamic equations to chains allows one to calculate the forces and moments acting on each bone and joint in the system, starting from the end links and moving to the beginning of the chain (the torso). For each chain we will apply Newton's laws of motion, and also take into account the transformation matrices described earlier.

For each bone *i* in the chain:

- Forces and moments:F_i– force acting on a bone, M_i– moment of force acting on the bone i;
- Mass and inertia:m_i– bone mass i, I_i– bone inertia tensor i;
- Angular and linear speed: v_i– linear velocity of the center of mass of the bone i, ω_i – angular velocity of the bone i;
- Angular and linear acceleration:a_i– linear acceleration of the center of mass of the bone, α_i– angular acceleration of bone;
- Kinetic equations: for linear motion $F_i = m_i a_i$, for corner $M_i = I_i \alpha_i + \omega_i \times (I_i \omega_i)$;
- Transfer of forces and moment between links: Force and moment acting on the bone$i$, bones are transferred $i-1$: $F_{i-1} = F_i + m_i a_i$, $M_{i-1} = M_i + r_{i,i-1} \times F_i$, where $r_{i,i-1}$– vector from bone $i$to bone $i-1$.

By applying these equations to each chain, starting at the end links, we can calculate the forces and moments for each bone and joint.

For the chain Torso – Neck – Head:

Transformation matrices (1), (2), cumulative matrix (3).

Linear equation of motion for the head (end link) $F_{head} = m_{head} a_{head}$.

Angular equation of motion for the head (end link):

$$M_{head} = I_{head} \alpha_{head} + \omega_{head} \times (I_{head} \omega_{head}).$$

Linear equation of motion for the neck $F_{neck} = F_{head} + m_{neck} a_{neck}$.

Angular equation of motion for the neck:

$$M_{neck} = M_{head} + r_{head,neck} \times F_{head} + I_{neck}\alpha_{neck} + \omega_{neck} \times (I_{neck}\omega_{neck}).$$

Linear equation of motion for the torso $F_{torso} = F_{neck} + m_{torso}a_{torso}$.

Angular equation of motion for the torso:

$$M_{torso} = M_{neck} + r_{neck,torso} \times F_{neck} + I_{torso}\alpha_{torso} + \omega_{torso} \times (I_{torso}\omega_{torso}).$$

The process is similar for other chains. For each chain, we determine the transformation matrices for each link, begin the calculation of forces and moments from the final link, applying the equations of motion and passing the results to the next link in the chain. We repeat the process until the initial link (torso) is reached.

By applying the dynamics equations sequentially to each bone and joint, starting at the end links and moving up to the torso, we can simulate realistic behavior of the entire system. These calculations allow you to take into account the forces and moments acting on each bone and joint, which is especially important for creating physically correct simulations in virtual environments.

4 Conclusion

Taking into account the laws of physics in mathematical rigging models based on determining the structure of the user's skeleton and objects, as well as constructing kinematics and dynamics equations, play a key role in virtual simulators for several reasons. Firstly, they provide realism of movement. Kinematics equations make it possible to describe the movement of objects, taking into account their positions, velocities and accelerations without taking into account the forces acting on them, which is important for creating realistic animations and modeling the trajectories of moving objects. Dynamic equations allow you to model physically reliable interactions between objects. Dynamics equations, including forces and moments, are important for simulating motion under external forces such as gravity, friction, and collisions, allowing you to model how objects will react to external forces, which is critical for creating realistic simulations of falling or colliding objects. In addition, kinematics and dynamics equations are important for motion control and control in virtual simulators. Controlling the movement of virtual characters in games and simulations requires precise control of their joints and limbs. Kinematics and dynamics equations provide the basis for such control, allowing characters to move naturally and react to interactions with the environment. These simulators allow students and professionals to practice skills in a safe virtual environment, simulating real-life physical conditions and object reactions. Virtual simulators allow engineers and designers to test and optimize designs and mechanisms in a virtual environment before they are physically implemented, saving time and resources during the development phase. These equations make it possible to analyze the stability, strength and efficiency of various systems and mechanisms, to identify and eliminate potential problems in the early stages of development.

References

1. B. Whitworth, *The Physical World as a Virtual Reality* (Massey University, Albay, Aukland, New Zealand, 2008).
2. I. Wohlgenannt, A. Simons, S. Stieglitz, Bus Inf Syst Eng **62(5)**, 455-461 (2020).
3. U. Uruthiralingam, P.M. Rea, Adv Exp Med Biol. **1235**, 89-101 (2020).
4. L.G. Sansone, R. Stanzani, M. Job, S. Battista, A. Signori, M. Testa, Virtual Reality **26**, 903-924 (2022).

5. A. Harichandran, J. Teizer, Automated Recognition of Hand Gestures for Crane Rigging using Data Gloves in Virtual Reality *In Proceedings of the International Symposium on Automation and Robotics in Construction*, 304-311 (2022).
6. A. Cellupica, Algorithms **17(1)**, 35 (2024).
7. W. Barfield, M.J. Blitz, *Research Handbook on the Law of Virtual and Augmented Reality* (Edward Elgar Publishing Ltd, 2018).
8. A. McStay, Philosophy & Technology **36(13)** (2023).
9. A. Zaporozhets, V. Artemchuk, *Systems, Decision and Control in Energy II* (Springer, 2021).
10. S. Pakarinen, J. Laarni, H. Koskinen, T. Passi, M. Liinasuo, T.-T. Salonen, Progress in Nuclear Energy **169**, 105104 (2024).
11. J.P. Canright, S.W. Brahmia, Phys. Rev. Phys. Educ. Res. **20**, 010146 (2024).
12. B. Whitworth, Res. Lett. Inf. Math. Sci. **11**, 44-60 (2007).
13. J. Pirker, E. Loria, S. Safikhani, A. Künz, S. Rosmann, Immersive virtual reality for virtual and digital twins: a literature review to identify state of the art and perspectives *Proc. of IEEE Conf. on Virtual Reality and 3D User Interfaces Abstracts and Workshops* (VRW), Christchurch, New Zealand, 114-115 (2022).
14. J.F. Lam, L. Gosselin, P.W. Rushton, Arch. Phys. Med. Rehabil. **99**, 2313-2341 (2018).