Simulation Module of Crank Rocker Mechanism Based on GTENG Engine

Weijun Chu*, Xiaohui He and Qing Liu
Army Engineering University of PLA, Nanjing, Jiangsu, 210007, China

*Corresponding author email: chuwj2008@qq.com

Abstract. GTENG engine technology is a complete set of real-time 3D engine that is mainly applied to the simulation field of engineering equipment. Crank rocker mechanism is common in engineering equipment. This paper analysed the evolution of crank slider mechanism, and two motion types and parameter compute of crank rocker mechanism. The design of CCrankRocker, associated mesh scene node and using coordinate system all were discussed. Further, compute way of motion simulation of the four components required for CCrankRocker was emphasized. The paper indicated that is needs set firm parent-child relationship between bar and slider when modeling in 3DS MAX, and enumerated some legends of the origin of each node and the directing of axis in coordinate system of crank rocker mechanism. Finally, the use of crank rocker mechanism module was illustrated by analysing the main function of Load function correlating model nodes with classes, and the motion simulation example of bucket rocker mechanism was given with partial program. The application shows that the simulation of crank rocker mechanism based on GTENG engine is smooth, practical and wide.

1. Introduction
3D virtual engine is widely used in civilian and military simulation field, such as game, city planning, virtual exhibition system, large-scale military simulation, flight simulation and so on. Many virtual engines are based on DirextX or OpenGL to drive hardware implementation, but the development workload for 3D simulation of using DirextX or OpenGL directly is huge, and the code reuse is low. So references 2 and 3 point out that simulation application developers can quickly realize the same functions by simply setting and programming the corresponding functions after encapsulating many reusable functions, thus they can greatly reduce the simulation project development difficulty and save the development time. References 2-5 speak of that there is no unified standard in the field of 3D visual engine. These engines have different understanding and definition of the function module, program structure, extended interface, model format and bottom implementation, so the degree of generalization of the geometry, elevation and physics models is low, and it is difficult to transplant each engine. Each engine has its own focus, which results in the insufficiency of the general engine and the limitation of the application scope of the specialized engine.

The self-developed GTENG engine technology is a complete set of real-time 3D engine, based on Windows operating system, and the underlying interface is for DirextX. DirextX is a multimedia control engine developed by Microsoft that runs on a computer platform (see reference 6). It is essentially a set of low-level application programming interface designed to provide a “hardware-independent” development platform for Windows-based applications. Its perfect compatibility with Windows gives DirextX a huge advantage over its peers. The framework of GTENG engine is shown in figure 1.
In the engineering equipment simulation system, the crank rocker mechanism is very common, but there is no literature to describe its dynamics and build a separate class of components on DirectX-based 3D engine. Therefore, this paper mainly focuses on the design of class of crank rocker mechanism (CCrankRocker) based on GTENG engine and the method of calling it to realize model simulation application. CCrankRocker is designed as a general module component, and when the corresponding model node and this class are invoked, the model can be used to complete the motion simulation of CCrankRocker and avoid repeated development, and improve the development efficiency of the project.

2. The Evolution of Crank Slider Mechanism

2.1. Four Evolution Forms of Crank Slider Mechanism

Crank slider mechanism is the most common type of mechanical simulation, as shown in figure 2.

The structure of crank slider mechanism includes three bars (a, b and c) and a slider, in addition, three angle values (A, B and C) are variable. The slider reciprocates in a fixed straight line. According to different fixed point (rack), crank slider mechanism can convert into four forms.

- **Crank slider mechanism.** Bar b is immobile, as shown in figure 2.
- **Guide bar mechanism.** Bar c is immobile, as shown in figure 3. If bar b can move in a circle, it is called rotary guide mechanism. If bar b can only swing, it is called swing guide mechanism. This change is actually due to the length of the guide rod.
- **Crank rocker mechanism.** Bar a is immobile, as shown in figure 4. The slider can only rotate around the C axis, while bar b slides. Lifting mechanism of a dump truck is a typical example of this structure.
2.2. Numerical Calculation of Crank Rocker Mechanism
Crank rocker mechanism is common in the field of engineering equipment simulation. Crank rocker mechanism in GTENG engine is a part of the dynamic engine, which is inherited from crank slider mechanism. Due to the symmetry of this device, it can be reduced to a flat mechanism whose dynamic characteristics be discussed by two dimension in dynamic simulation.

No matter which variant it is, the lengths of bar a and c are invariable, while the length of bar b is variable as the slider moves on it, and the angles of A, B and C also change accordingly. In practice, the lengths of a and c are known and constant, the three initial angles of A, B and C are known, and the initial value of the length of bar b is known. There are two motion types:

- To select an initiative-revolving bar, when the bar rotates at an Angle \( \theta \), the length of bar b changes, as well as the three angles are change. So all variational value needs to be computed to confirm the final state.
- The length of bar b firstly changes to result in the rotation of other bars.

The end position equation of crank rocker mechanism was deduced in reference 7, and the motion simulation model of the mechanism was established. Through the kinematics simulation of Pro/e mechanism, the trajectory of the motion process is verified.

As shown in figure 4, the position equation of crank rocker mechanism is

\[
\begin{align*}
ccos\theta - a &= b\cos\delta \\
\frac{c\sin\theta}{b} &= \sin\delta
\end{align*}
\]

So \( \delta \) and b is

\[
\begin{align*}
\delta &= \arctan2(\frac{asin\theta}{(acos\theta - a)}) \\
b &= \sqrt{(ccos\theta - a)^2 + c^2\sin^2\theta}
\end{align*}
\]

3. The Simulation Method of Crank Rocker Mechanism Based on GTENG Engine
3.1. CCrankRocker
Reference 8 designed a base class based on Directx called DreamBehavior, in which several virtual functions were declared, such as Update function, which was called once per frame. The user then declares a new class that inherits DreamBehavior base class and rewrites the functions, and finally binds the new class to the corresponding object as a component, allowing the user to customize the functional component for more functionality.

CCrankRocker was inherited from CRod. When using this class, it must be associated with mesh scene node, with each bar and slider corresponding to one node. For convenience, the local coordinate origin of the node is positioned for making the model.
In *CCrankRocker*, move is along x-axis direction which positive is forward and negative is backward. It rotates around y-axis. GTENG engine uses a left-handed coordinate system, while general modeling software, such as 3DS MAX, uses a right-handed coordinate system, with z-axis corresponding to y-axis in this engine. *CCrankRocker* needs a common coordinate space, and all components of *CCrankRocker* will be converted to this space to calculate variables such as position, length, and angle. Otherwise, each component has its own local coordinates, and points in local coordinates and points in another local coordinates do not mean the same point even if they have the same coordinates.

3.2. Four Components Required for *CCrankRocker*

Four components are bar a, b, c and slider, as shown in figure 4. The length of bar a (or BC) is variational, and slider is sliding on bar a. GTENG engine represents these components in terms of nodes, each node having only one origin of coordinates, which can be easily obtained. For example, the origin of bar c is at point A, the origin of bar a is chosen at point B, and the origin of slider is at point C. The lengths of the three bars need to be calculated before you can figure out the angles of the three angles. The length of bar a is calculated by the distance between the origin of slider and the origin of bar a. Bar c is a bit of a problem, because only the origin is known, and the other end point B is not defined at the node. Although we know that the origin of bar a coincides with point B, we cannot assume that the other end of bar c necessarily coincides unless we have rotated bar c correctly to make it coincide. Therefore, an auxiliary node needs to be located the origin of this node at the point B of bar c. This is what *m_cRodAssist* members do.

3.3. Node Relationship

The parent-child relationship between bar and slider needs to be set firm relationship when modeling in 3DS MAX, and the parent-child relationship will not be changed every time when modeling. The required node relationship is shown in figure 6.

![Figure 6](image_url)

The origin of each node and the directing of axis are shown in the following table.

| The parent node shared by hinges | Hinge 1 (C) | Hinge 2 (A) |
|----------------------------------|-------------|-------------|
| Slider                           | Bar c       | Auxiliary node of bar c |
| Bar a                            |             |             |
Table 1. Coordinate system of crank rocker mechanism.

| Node          | Legend | Origin position | The directing of axis x | The directing of axis y | The directing of axis z |
|---------------|--------|-----------------|--------------------------|-------------------------|-------------------------|
| Slider        | Point C | Any direction in the plane | Perpendicular to the mechanism plane, outward | Any direction in the plane |
| Bar b         | Point A | In the direction of AC, C goes to A is positive and negative conversely | Perpendicular to the mechanism plane, outward | Any direction in the plane |
| Bar c         | Point B | Any direction in the plane | Perpendicular to the mechanism plane, inward | Any direction in the plane |
| Auxiliary node of bar c | Point A | Any direction in the plane | Perpendicular to the mechanism plane | Any direction in the plane |

Note: Using a left-handed coordinate system in above table, 3DS MAX uses a right-handed coordinate system where z-axis and y-axis are switched and three axes directions are invariant.

If the mechanism operation is not correct when running the program:
- To check whether the coordinate axis direction is correct and adjust the coordinate axis direction of the parts until the mechanism operates normally;
- To check whether the coordinate origin of each part is correct and adjust the coordinate origin position of the part until the mechanism operates normally.

These are the most common mistakes.

4. Use of Crank Rocker Mechanism Module
To use the crank rocker mechanism, we need inherit a class from CCrankRocker and implement the definition of pure virtual function (Load). We can do actual simulation only after having associated the model with CCrankRocker. The main function of Load function is to correlate model nodes with classes, as follows:

```cpp
class CMyClass : public CCrankRocker
{
public:
    CMyClass();
    virtual ~CMyClass();
    virtual bool Load (CBaseSceneNode* pRoot);
};
```

Load function has a unique parameter, a pointer to the root node which can be used to search the pointer of child node by the node name.

```cpp
m_pNode = pRoot->FindSceneNodeByUnicode("The node name to find");
```

In order to associate the model with crank rocker mechanism module, at least six nodes with parent-child relationship are required, such as:
m_aRod= pRoot->FindSceneNodeByUnicode("The hydraulic cylinder");
m_slider= pRoot->FindSceneNodeByUnicode("The hydraulic rod");
m_cRod= pRoot->FindSceneNodeByUnicode("Driven rod");
m_cRodAssist= pRoot->FindSceneNodeByUnicode("Driven rod end");
m_hinge1= pRoot->FindSceneNodeByUnicode("Hydraulic cylinder support");
m_hinge2= pRoot->FindSceneNodeByUnicode("Driven rod support");

By associating these nodes, we can call Step function to make crank rocker mechanism move, as shown in figure 7.

Figure 7. Two states of motion simulation of bucket rocker mechanism

5. Conclusion

- We create CCrankRocker based on GTENG engine, specify its coordinate space, define its four components, and determine their node relationships. The function of connecting model node with CCrankRocker is given, and the motion simulation of crank rocker mechanism model in virtual simulation system of engineering equipment is realized by calling Step function.

- The motion simulation of crank rocker mechanism shows that the program runs efficiently, the executable program is small, the motion of the mechanism is smooth and real. The application of crank rocker mechanism module based on GTENG engine can avoid repetitive work, improve development efficiency, also can be widely used in military, education, film, game and other fields.

- By introducing the idea of virtual engine development, it can provide an idea and approach for studying the motion simulation of other kinds of mechanism.

References

[1] Chu W J, He X H, Zhu Z C 2019 Research on the Semi-physical Simulation Maintenance Training System of One Engineering Equipment Advances in Intelligent Systems Research and Engineering vol 168, pp 34–35

[2] Zhao W, Mao P, Shen F Y 2019 Development Trend of Next Generation 3D Graphics Engines Journal of System Simulation vol 29, pp 2935–2942

[3] Ren S Q 2014 Research and Implementation on Key Technology of 3D Graphic Engine (Wuhan: Huazhong University of Science and Technology) pp 44–51

[4] Liu C 2019 Development of Panoramic Video Play System Based on Real-time Engine Advanced Motion Picture Technology vol 168, p 17

[5] Guo L W, Wang Q, Gao K B, Gu W 2019 Research on Armored Vehicle Motion Simulation Based on Virtual Reality Computer Systems & Applications pp 221–227

[6] Hu S D, Zhao C L, Li C L, Fan Z J, Chen H J 2012 Development of DirectX Visualisation Simulation System Oriented to Virtual Training Computer Applications and Software vol 29, pp 129–132

[7] Liu B H, Wei M, Tao Y 2018 Design and Simulation of Capping Machine Based on Crank Rocker Mechanism Packaging Engineering vol 39, pp 159–161

[8] Qin X J, Wang B, Dong J J, Zhan L C 2017 Framework and Interface Design of Game Engine Based on DirectX Library Digital Technology & Application, pp 182-185