Chrono::R3D

User manual

Contents

1 Introduction

Welcome to Chrono::R3D , the complete multi-body 3D simulation software.

Chrono::R3D allows you to perform simulations of virtual mechanism on your computer, made of parts, actuators, motors, links between parts, spring, dampers, etc. For example you can build a car, turn the engine on, drive it and watch the road handling during manouvers: the results are physically correct. The car behaves like it were made of real tyres, steering arms, springs, suspensions.

You can use Chrono::R3D to simulate a wide set of mechanisms, such as cars, robots, trucks, trains, car suspensions, earth-moving machines, motorscrapers, backhoe loaders, human skeletons, aerial vehicles, landing gears, robotic manipulators, engines, torque converters, prosthetic devices, artificial arms, miniaturized mechanisms, and so on..

The Chrono::R3D plugin is an add-on module for the Realsoft3D software. This plugin is based on the Chrono::Engine C++ programming library for developing applications about physical simulation.

1.1 Chrono::R3D highlights

Chrono::R3D is a full featured multibody simulation package, based on advanced researches in theoretical mechanics and applied math. Engineering, research and machine design can take advantage of the reliability, quality and complexity of simulations; in the meanwhile the interface is so intuitive and fast that also 3d animators and artists can use Chrono::R3D in film industry, for physically-based animations, advanced character animation and so on. This means that either if you are an engineer, an expert of biomechanics, a 3d animator, a game designer, an accident reconstructionist or a scientist, you will find lot of useful features in Chrono::R3D .

- Chrono::R3D provides the time-plots of all variables of your system (position, speed, acceleration, forces, etc.)
- Both inverse-kinematics and dynamical analysis can be performed.
- Chrono::R3D allows static analysis, with geometric and structural non-linearities.
- There is a wide set of constraints: revolute, ball joint, rack-pin, cylindrical, gears, hook, and much others. Constraints have the most advanced non-linear settings to simulate non-linear behaviours (link limits, non-linear damping and stiffness, impulsive restitution, limits with ”cushioning”, internal forces, rheonomic forces, etc.)
- Chrono::R3D offers a wide variety of integration settings, for math-skilled users which need the highest precision in results.
- Forces and torques can be applied everywhere, even time- and position-varying. Also spring-dampers systems can be applied. Positional and force actuators can be placed where you need them.
- Chrono::R3D finds all the reactions in the constraints of your mechanism (with time-graphs) when performing the dynamical simulation. You can discover if your device is well designed before building it.

- Complete simulation of large non-linear movements of your devices, with an interactive 3D animation of all the motions. The user can modify the applied forces and constraints even while the simulation is running.
- Chrono::R3D allow ”virtual prototyping” at its best. The animations can be recorded, reviewed, fine tuned, modified by key-framing, saved for WEB publishing or teamwork, rendered on videotapes for presentations with true photorealistic quality. When we say true photorealistic quality, we do not mean simple OpenGL previews, like in other products, or platic-like scanline pictures; we mean true raytracing with the power of Realsoft3D multithreaded rendering engine, with advanced features like volumetric materials, programmable shaders, complex optical effects, etc. All available in the same environment.
- Complex mechanisms can be arranged into parts and subparts, with hierarchical organization of parts.
- Chrono::R3D fits seamlessly in the Realsoft3D environment. The plugin allow you to apply constraints between objects, directly in the 3d editor. For example you can build a robot arm, set the movement of the manipulator, and let Chrono::R3D simulate the entire motion with inverse kinematics.
- Also dynamic simulation is available: build a car, apply a strong torque to the wheels, and watch the car do a ”power slide”... And if it goes off-road, the suspensions will react accordingly, with bounces and vibrations.
- Also build pendulums, people doing base-jumping, engines, bicycles, etc.
- Collision detection is supported. Collisions are detected between complex compound shapes in few milliseconds: in many cases, the overhead of the collision detection is so low that it allows real-time simulations of scenes with dozens of bodies.
- Friction between colliding objects is computed using a state-of-the-art LCP/NCP iterative solver, aiming at high computational speed and robustness.
- Thank to Realsoft3D ”model-view” concept, the modifications of constraints, forces and objects are instantly displayed in the 3D views, and the simulation can be modified interactively, with a ”man in the loop” approach.

1.2 Main features

Chrono::R3D is a multibody system based on lagrangian dynamics, where differential equations of motion are expressed with natural coordinates (position and speed of bodies, either cartesian xyz and rotational). The constraint between bodies are expressed through additional algebraic equations. Therefore, Chrono::R3D has to solve a system of mixed differential-algebraic equations (DAE) while the integration proceeds; in case there are contacts with friction, the problem is even more complex and becomes a differential-complementarity-problem (DCP). Hypercomplex numbers (quaternions) are used internally as rotational coordinates; this avoids singularities and the simulations perform better (the user can choose between different angle sets for more user-friendly rotational coordinates).

A Chrono::R3D user can model a mechanical system this way: the mechanism is basically made of ”bodies” (objects), each body can contain ”markers”, and ”links” can be entered to constrain the relative movements of markers. Theres no need to identify kinematical chains, like in other software, so you wont ever worry about closed-loops chains, branching of chains and so on, you just create constraints between bodies.

Here is a list of most important features of Chrono::R3D .

1.2.1 Body features

- Mass
- Inertia (diagonal and mixed)
- Position
- Alignment (different rotational coordinates can be used to enter data or to output simulation results)
- Speed
- Angular speed
- Collision detection properties (static/dynamic friction, restitution coefficients, etc.)

1.2.2 Marker features

- Body-relative position
- Body-relative alignment (quaternions, Cardanos, Eulero, etc.)
- Body-relative speed
- Body-relative angular speed
- Body-relative acceleration
- Body-relative angular acceleration
- Absolute position
- Absolute alignment
- Absolute speed
- Absolute angular speed
- Absolute acceleration
- Absolute angular acceleration
- Imposed motion of marker about body (X-Y-Z-Angle motion functions, with user-friendly functions such as imposed speed, ramp, sine wave, motion capture, polynomial function, hand-drawn curve, sigma ramp, formula, etc.)

1.2.3 Force features

- body-relative position
- body relative direction
- absolute position
- absolute direction
- mode: force or torque
- functions to set a non-linear behaviour (for example to have a force which smoothly vanishes as the time passes. Here the user can play with speed fx, ramp, sine wave, motion-capture streams, hand-drawn curves, formula, etc.).

1.2.4 Link features

- Many constraint types, for example:
  - ⋆ free
  - ⋆ locked (welding),
  - ⋆ spherical,
  - ⋆ cylindrical,
  - ⋆ revolute,
  - ⋆ prismatic,
  - ⋆ point on plane / on line,

  - ⋆ Oldham joint,
  - ⋆ homocinetical joint,
  - ⋆ Cardano shaft,
  - ⋆ hook,
  - ⋆ screw,
  - ⋆ gears,
  - ⋆ alignment,
  - ⋆ parallel,
  - ⋆ perpendicular,
  - ⋆ point on spline,
  - ⋆ point on surface,
  - ⋆ trajectory,
  - ⋆ etc...

- Each link has a mnemonic 3d picture and a custom wireframe representation.
- Link imposed motion (to build actuators, imposing the relative motion of two bodies. You can use X-Y-Z-Angle motion functions, with user friendly GUI for functions such as imposed speed, ramp, sine wave, motion capture, polynomial function, hand-drawn curve, sigma ramp, formula, etc.)
- Constraint violations
- Link relative coordinates (position, rotation, speed, angular speed, acceleration, angular acceleration)
- Link internal imposed forces (for each degree of freedom, the user can set a force, a spring and a damper. These parameters can be nonlinear in space and in time. For example, one can draw a graph of the torque acting on the degree of freedom Rx, to simulate an electrical engine which rotates a wheel, with time-dependant power. Or a prismatic link can have a nonlinear spring along the Z axis to simulate a Mac-Pherson suspension in a car simulation, where the spring becomes very stiff when the end of the guide is reached, and so on..).
- Link limits (the user can set upper-lower limits for all the degrees of freedom of the link. Moreover, each limit can have a ”proximity cushion”, with stiffness and damping. Both stiffness and damping of the proximity cushions can be non-linear, using one of the many available functions -read above-.
- Specialized links to build engines, linear actuators, brakes etc. For example, the user can set torque-speed diagrams for engine constraints.

1.2.5 Special constraints

- Imposed trajectory, to make a point follow a parametric NURBS curve
- Special link objects to make gears (internal, external, conical)
- Spring-damper systems, with non-linear properties,
- Brakes and clutches, with stick-slip effects,
- etc.

1.2.6 Actuators

- Motor actuators, with imposed speed or imposed rotation or imposed torque. The user can modify the torque-speed curves or apply specific motion laws. A reducer can be applied, with specific transmission ratio and efficiency. The motor can ‘learn’ the motion, when switched off.
- Linear actuators, with user imposed motion or force. The actuator can ‘learn’ the motion, when switched off.

- Pneumatic linear actuators. The user can set pressure, valve opening, stroke length, diameter, etc. Effects caused by air compression and outlet conductance are taken into account.

1.2.7 Scripting

- Scripting via Javascript programs.
- Most variables of Chrono::R3D objects can be accessed and read or modified via scripting.
- Scripting can be executed in a shell or introduced with programs.
- A special object ‘controls’ can be used to build mechanisms with embedded scripts, to be executed at each simulation step. This allow the design of automated devices, AI machines, etc.
- The ‘controls’ object can be used also to assign a graphical user interface, a ‘panel’, to a device. The user can move sliders on this panel during the simulation, in order to control some variables (ex. drive a car in realtime simulations)
- A special Javascript object can be used to simulate PID digital controllers for the design of mechatronic devices with feedback circuits.

1.2.8 Optimization

- Chrono::R3D embeds an optimization tool which can be used to find automatically the best design for your mechanisms. It uses Javascript functions, but an easy graphical interface is provided.
- Optimization engine based on applied genetic evolutionary theory, for global finding.
- Optimization engine based on gradient search engine, for local refinement.

1.2.9 Data recording

- Record variables into x-y functions. These can be saved in Chrono::R3D .chf format, or in ASCII streams, or in .eps plots, or in Matlab format.
- Plot trajectories of whatever point of moving parts.
- Plot x-y diagrams onto curved surfaces.
- Draw 3D vectors, to show instant speed of points, or to show reactions in constrsaints, etc.
- Record whatever Javascript variable, or plot formulae which use variables.

1.2.10 Simulation features

- Inverse Kinematics. The IK can be run as an animation, or it can be performed interactively by the user (for example the user turns a wheel with the mouse in the 3d view and a mechanism gets the movement).
- System assembly. Redundant or misplaced links are automatically removed.
- Full multibody dynamical simulation.
- Fast collision detection during simulations.
- Non-linear static analysis.
- Environmental forces (es: gravity)

Note that all the variables of Chrono::R3D can be recorded into graphs as the simulation proceeds. Graphs can be edited, fine-tuned, filtered and printed if necessary.

The Chrono::R3D plugin can use the advanced features of Realsoft3D , for example the macro recording, the model-view relationship, the interactive modeling, the control through external programs.

Of course, together with all these features the user will get the power of default Realsoft3D , such as the advanced visualization methods, the photorealistic rendering, the NURBS and the CSG modeling, the advanced animation system, the shading system, etc.

1.3 Differences between the licenses

Three license types are currently available for Chrono::R3D :

- Chrono::R3D STUDIO, including most relevant features, targeted at animators, students, etc.
- Chrono::R3D EXTREME, the complete toolset, and targeted at engineers and researchers,
- Chrono::R3D CUSTOM, like Chrono::R3D EXTREME, but includes also a developer license of Chrono::Engine .

That is, the difference between the EXTREME and the less expensive STUDIO license is that some features are disabled; in detail the STUDIO license does not allow the following functions:

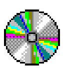
- Saving ChFunctions (as files of type .CHF).
- Saving PostScript EPS plots of recorded variables.
- Analysis of ChFunctions (mean, min max, etc.)
- Design tool which computes and plots the jacobian for robots.
- Design tool which performs global and local optimizations.
- Design tool which shows the working volume of a robot.

1.4 How to use this manual

Before using Chrono::R3D , follow the guidelines in the installation chapter.

The manual is splitted in three parts: the **user manual** section, with tutorials, the **reference** section, and the the **scripting** section.

After you have installed the software, we strongly encourage you to start with the tutorials chapter, repeating the examples on your computer, in the sequence they are presented. This is the fastest and easiest way to understand the concepts and the philosophy behind Chrono::R3D .



This symbol means that there is an example on disc.

This symbol is shown aside helpful tips and hints.

This symbol stresses the importance of a concept.

2 Installation

Requirements:

- Requirements
- Pentium 1GHz or higher (dynamical simulations require a lot of processing power, so the fastest the processor is, the fastest the animations will be computed, even in real-time). Pentium 3GHz recommended.
- 256 Mb Ram (1Gb+ highly recommended)
- Windows 2000 / Windows XP / Windows ME / Windows Vista
- The software Realsoft3D V5 must be already correctly installed into your system. Installing also the Service Pack SP4 for Realsoft3D V5 is strongly reccomended.

Procedure:

- **Insert the disc into the drive and double click on the Setup.exe icon.**
- **Follow the instructions on the screen until the installation procedure will be completed.**
- **That's all. After the plugin has been installed into your system, it will be automatically loaded all times you start Realsoft3D .**

The first time you will start Realsoft3D with Chrono::R3D installed, you will see a splash screen with the 'Chrono::R3D logo' and a request: do you want Chrono::R3D to add its tool buttons to your tooltab? Leave the checkmark ON, and all Chrono::R3D tools will be automatically added to your tooltab. Switch OFF this checkmark only if you want to add Chrono::R3D buttons to the interface by yourself.

If you did not let the splash screen to install the Chrono::R3D tools in your environment, each time you will start Realsoft3D you wont find any new icon in the tool bar, or any new menu. This is because Chrono::R3D installed some new tools into Realsoft3D , but it may be up to you to decide when and where put the tool icons in the toolbar.

To add Chrono::R3D icons in the toolbar by yourself, without using the automatic configuration of the interface, do the following simple steps.

- In Realsoft3D execute the menu Customize/Available objects. This will pop up a window with all the UI (user interface) elements of Realsoft3D .
- Press the tool icon. A list of tabs containing lot of tool icons will be shown. If Chrono::R3D has been correctly installed, among these tab theres the Chrono tools tab. Switch to that tab.
- Drag and drop all the tool icons from the Chrono tools tab to the current tool tab, so the the Chrono::R3D tools will be always available, just like the default move, select, rectangle etc. tools.
- To make the UI configuration permanent you should save the changes. Use project/save startup. Now the Chrono::R3D tool icons will be shown on the toolbar each time you start Realsoft3D .

3 Tutorials

Before starting the tutorials:

- be sure that you already know the basic features of the Realsoft3D interface (ex: using views, tools, grids, select window, property window, etc.) If not, please spend some time to experiment with Realsoft3D , reading its manual.
- check that your Chrono plugin has been correctly installed (follow the guidelines of the previous chapter). If so, you should be able to access the following Chrono-tools on the toolbar:

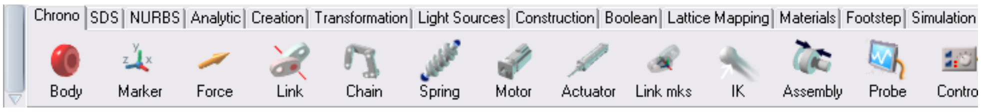


Figure 3.1 Toolbar with Chrono::R3D tools

If all is fine, you can begin with the first tutorial.

3.1 Tutorial: create a link

LEARN

creation of constraints between body and ground
hierarchic structure of Chrono::R3D items
grounded objects
the assembly tool

In this tutorial you will use the LINK tool, a very important tool from the Chrono::R3D toolset, which creates constraints between the parts of your mechanism. Please read carefully this tutorial and the notes, because here you will learn some of the most important concepts about the way Chrono::R3D works.

1

Go to ”front” or ”side” view.

2

Create a vertical rectangle with one of the modeling tools of Realsoft3D (for example use the tool ”Rectangle” from the Analytic toolbar). This will represent the rough shape of a pendulum.

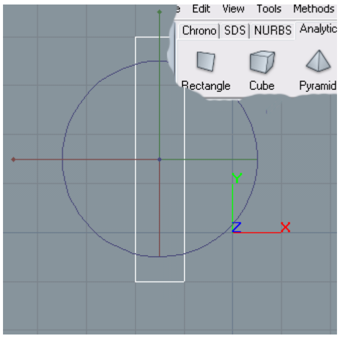


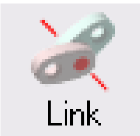
Figure 3.2

3

Select the rectangle (from select window or using the ”select” tool and mouse pointing). Maybe its already selected after you have created it.

4

Select the ”Link” Chrono-tool (it will be used to create the joint between the rectangle and the world). Note the instructions on the status bar, on the bottom of the screen.



5

Now the Link tool wants to know the second object for the constraint: just click on an empty region of the 3D view (the ’air’) to tell that you want to link the rectangle to the world.

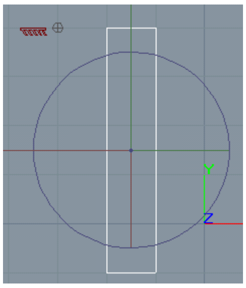


Figure 3.3

6

The Link tool also wants to know the point where you want to put the joint, so double-click near the left end of the rectangle: a spherical joint between the pendulum and the ground absolute reference will be immediatly created.

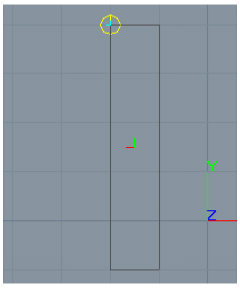


Figure 3.4

7

Well, the link creation has automatically set up your basic multibody system: just a rigid body, a link and some other Chrono objects. You could already click on the play icon of the animation window.

Before going on with the tutorial, stop a minute and look at the select window. You can see that the operation of creating a link has changed some things in the object hierarchy.

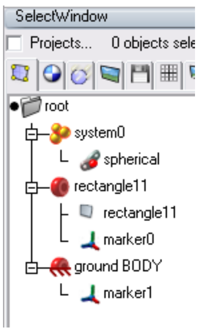


Figure 3.5

A ”System” object has been created: this object represents the settings of the simulation engine. This object is invisible in view windows, but it is the ”brain” of all Chrono::R3D animations. Some rules: 1. If system objects are missing in our hierachy, they are created automatically by Chrono when needed (like in this tutorial) at the root of your working level. 2. If needed, you can create system objects by hand, simply using the popup menu of select window ”create/Chrono system”. 3. You can move a system object where you want in the hierarchy. 4. If you add more than one system (for example for parallel decoupled simulations or other strange applications), the settings of the last one will be taken into account.

The rectangle has been enclosed into a ”body” object. Such kind of Chrono::R3D object represents a rigid part (i.e. a ”rigid body”). All rigid bodies have a mass, an inertia and a center of mass associated with them, default values are used in this tutorial, but you can edit them -more on this later-. Chrono simulation engine actually affects and moves these ”body” objects. For visualization purposes, you can put whatever geometric primitives, materials and surfaces into such ”body” object, i.e. into its sub-hierarchy. They will be moved with it, representing a single rigid body. In this tutorial you did not worry about creating a ”body” object because the ”link” tool actually created one for you, but you could use the ”body” tool icon and create an empty rigid body (only its inertial coordsystem will be shown in view window) and you could fill it with geometric primitives later.

A ”ground” body object has been created automatically. This is a ”body” object too, just like the rectangular pendulum, but it is fixed to the absolute reference (the world, or say it the ”ground”). It has no geometric shapes in its sub hierarchy. It is needed only because you created a link, i.e. the spherical joint, between the link and the ground.

A ”Link” object has been created into systems sub-hierarchy. Link objects are very important because they represent the mechanical costraints between rigid bodies and in fact they define the mechanical system. By default they are put into the sub level of the system object, which is a comfortable place to put them, but you could put them in whatever other part of the hierarchy.

Two ”marker” objects has been created, one into the ”ground” body and one into the pendulum body. This happened because you created a link between ground and the pendulum, and that link must know ”which point of body A is constrained to which point of body B”. So the marker objects are these ”points”, that is the body-relative coordsystems which are used as references by links. A marker must always belong to a body hierarchy: it has no meaning if its outside. You can create your own custom markers with the ”marker” tool in the toolbar.

Normally you dont have to worry about markers, systems and so on, because the ”link” tool (the one you will use most often) will set up these objects for you. However it is better to know their meaning in order to understand how Chrono::R3D works, and maybe to do troubleshooting if your animation does not work as expected.

7

Now try to select the pendulum object and to move it apart: you can use whatever Realsoft3D tool to modify it (for example, use the Move and Rotate tools of the ”Transformation” toolbar, or drag object’s handles). Note how a red line indicates that a link has been ’dismounted’.

8

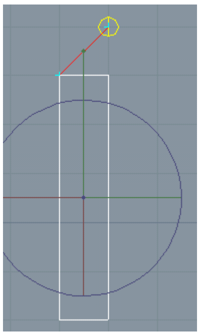


Figure 3.6

Click on the Assembly tool: note how the mechanism will be mounted agin, in order to satisfy all links.

9

Use the popup menu (right mouse button on the 3D view) to open the ”property window”. When the rigid bodies are selected, you can find their properties in that window, under the ”Spec” tab.

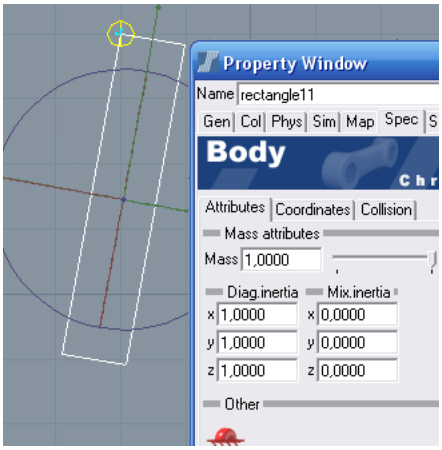


Figure 3.7

3.2 Tutorial: links between bodies

LEARN

creation of links between bodies
modify references for link position
change link type
marker absolute and relative position

In this tutorial you will use again the LINK tool, and you will create a double pendulum.

1

Create two rectangles, as shown in figure (for example use the tool ”Rectangle” from the Analytic toolbar). This will represent the two rigid bodies which define your double pendulum.

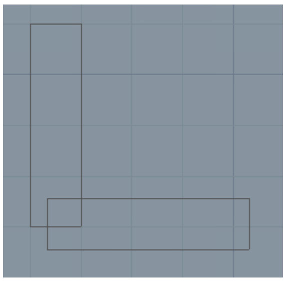


Figure 3.8

2

Unselect all objects (for example, click on a blank area of the select window, or drag the mouse in an empty area of the 3D window).

3

Click on the ”Link” tool of the ”Chrono” toolbar: you must perform 4 steps.

4

Step1. Since in this example we started with no object selected, the Link tool now wants to know which is the first object to constraint (see the help text in the status bar, at the bottom of the screen!). Therefore, we move the mouse cursor over the vertical rectangle and we click to select it as ”body 1” -note that it changes colour when the mouse passes over it, to help you understand what you are going to select-

5

Step2. You have to select the second object (”body 2”), which will be constraied to the body 1. Simply go on the horizontal rectangle and click.

Note that, during Step2, a symbol with ////// slashes appeared near the cursor when you moved it on empty areas of the view: this means that, if you had clicked, you would have constrained body 1 to the ground, instead of constraining it to body 2. As in first tutorial.

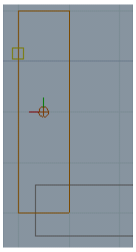


Figure 3.9

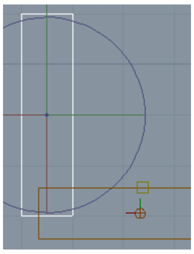


Figure 3.10

During steps 1 and 2 of link creation, turning OFF the snap-to-grid feature of Realsoft is strongly suggested, because grid snapping makes objects selection more difficult.

6

Step3 and Step4. To complete link creation, click twice on the position where you want to create the spherical joint between the two rods. The constraint is created.

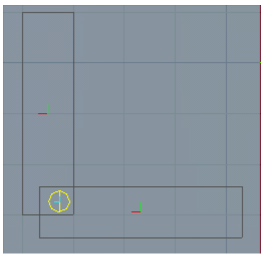


Figure 3.11

Why click twice to conclude Step3 and Step4 of link creation? In fact, the first click tells the position of the joint respect to body 1, and the second tells the position of the joint respect to body 2 -if the mechanism is already mounted, a fast double click on the same position is enough-.

7

Now you will create a spherical joint between the vertical rod and the ”ground”. Again, use the Link tool. If you learned Tutorial 1, this is easy to do.

8

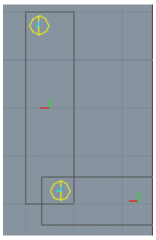


Figure 3.12

Last, you will create a third link, between the horizontal rod and the ”ground”. Again, use the Link tool like before, but this time we want to make a different type of joint. In fact we will choose ”point-line” as Type, in the toolbar option.

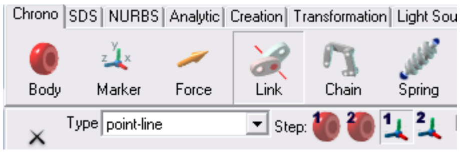


Figure 3.13

9

This done, the third created link represents a constraint where a point of the horizontal rod must slide on the yellow line.

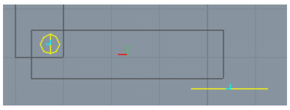


Figure 3.14

10

What if you needed a different direction for the yellow line of the point-on-line constraint? Go to the Select window, and select the second ”Marker” object in the Ground sublevel, rotate and move it a bit. You can move and rotate also other markers. Click on the Assembly tool.

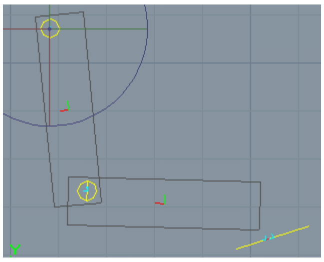


Figure 3.15

Each "Link" object uses two Marker objects, belonging to two different Body objects, because these markers are used as auxiliary coordinate systems for building the constraint (they show the place where the two bodies must be joined). Hence you can freely modify the position of markers, to change the position of links.

3.3 Tutorial: links between bodies

LEARN

interactive inverse kinematics
change link type

In this tutorial you will use the mouse to move a mechanism, with the help of the IK (Interactive Kinematics) tool. The IK tool can be used to move parts of whatever mechanism with some degrees of freedom: this means that we can pick some of the bodies and drag them with the mouse, to understand the admissible motions. Note that we suggest you to complete Tutorial 2 before reading this, because we will start from that example.

1

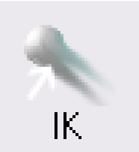
First, create a mechanism like the double pendulum of the previus tutorial.

2

Select the rigid body of lower horizontal rod (either in select window or by mouse picking in 3d view window).

3

Click on the "IK" tool of the "Chrono" tooltab, hence activating the Interactive Kinematics feature.



4

Click in the 3D view, near the horizontal rod. Imagine this is the picking point (as if your hand were grabbing the selected rigid body in that point).

5

Move the mouse, and see that the whole mechanism follows the pointer.

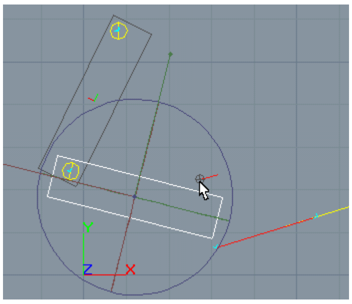


Figure 3.16

6

Click again to accept and finish the IK tool. The body will be unselected.

If you start the IK tool with no rigid body selected, you will use the first click in the 3D view to select it, then you can move it as usual.

7

You could repeat these steps to perform other interactive kinematic tests with the IK tool, maybe selecting the other rigid body (the vertical rectangle).

8

Try also using the IK tool on a different input plane. For example, go in perspective mode, set the YZ input plane, use the IK and see that the mechanism is dragged out from the front plane where it was created (in all cases, the IK picking point is always projected onto the current input plane, and the dragging happens there).

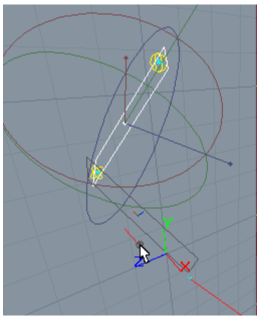


Figure 3.17

9

The IK tool does not act only on rigid bodies. In fact, you can use it also on the markers contained in the ground level: try to select one of the two markers in the ”ground” sublevel and use IK, you will see that the reference to links will change in realtime.

If you are not satisfied with the result of the interactive motion, instead of doing the second click -which would accept the result- you could simply press the right mouse button and choose ”Cancel” in the poput menu.

10

The double pendulum contains two spherical joints as links, and this explains why it is possible to move it out from the front plane where it was created. Anyway, in case you want to change the type of these links, try to do the following: select the two spherical links in the select window (in the System sublevel) and use the popup menu ”Properties...”. In the property window, go to the Spec tab, which shows the properties of the links. Here you can change the link type from ”spherical” to ”revolute” -by the way, see how the small picture changes-.

11

Click on the ”Assembly” tool of the Chrono tooltab, or use the IK tool, and you will instantly see the effect of the different link types.

Figure 3.18

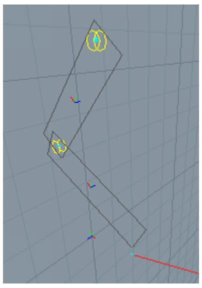


Figure 3.19

Each Chrono object has custom settings in the Property window. In the Spec tab of that window you can access and fine tune the most important settings. For example, try to select a Marker object and look at its property window: here you have the numeric fields to enter its absolute or body-relative position.

3.4 Tutorial: add springs and dampers

LEARN

spring-damper links
change custom settings of spring
delete links

In this tutorial you will add springs to a mechanism. You will start from the object which you created in the last tutorial.

1

First, create the mechanism described in the previus tutorial (a sort of double-pendulum).

2

We want the lower end of the double pendulum to be free: select the ”point-line” link in the System sublevel and delete it.

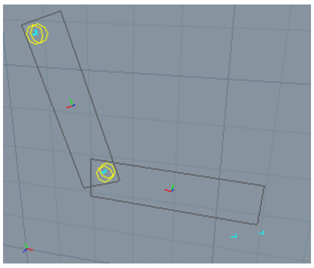


Figure 3.20

When you delete a Link object, the markers that were used as references are left -they are not deleted automatically. Later, you may delete these marker object one by one, if you want.

When you delete a Marker object which were used as a reference for a link, such link is not deleted automatically (it simply becomes ”not active”).

3

Be sure nothing is selected.

4

Click on the ”Spring” tool of the ”Chrono” tooltab (it works almost in the same way of the ”Link” tool seen in previous tutorials).

5

First step of the spring tool: since no object is initially selected, you must select the first body to constrain. Simply select the lower rectangle as body 1, by picking it with the mouse.

6

Second step of the spring tool: select the other object to constrain. Since here we want to link body 1 (the rectangle) to the world, you can click on an empty region of the view window, where the ///// symbol near the cursor tells that the ”ground” will be used as body 2.

7

Third and fourth steps of the spring tool: click at the beginning and at the ending of the spring. (the beginning may be near the rectangle, that is body 1, and the ending may be a bit above, as the spring were joined to an invisible hook).

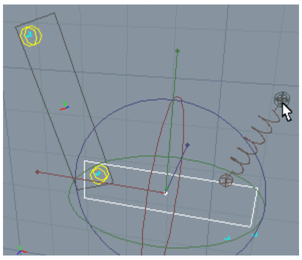


Figure 3.21

8

Ok, the spring is now created. See in select window how a new link object of type ”spring” is added in the System sublevel.

9

We want to create another spring, this time between the two rectangles: click on the ”Spring” tool. (or use the RMB to show the popup menu in view window and choose ”Again”).

10

Perform the four steps of link creation, picking the two rectangles in steps 1 and 2, and clicking on spring beginning and spring ending points for steps 3 and 4.

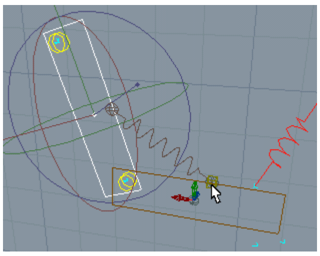


Figure 3.22

11

Use the IK tool to move the pendulum, to see how the springs are stretched as the parts move.

12

You can modify the stiffness of the springs (and many other parameters) by opening the property window of the spring link. Go to the ”Spec” tab and in the ”Custom” sub tab to find the parameters.

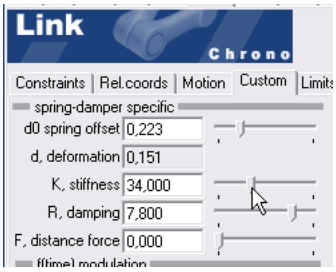


Figure 3.23

13

Ok, the pendulum is ready for dynamical simulation, explained in the next tutorial.

3.5 Tutorial: simulation (dynamics)

LEARN

dynamics simulation
rewind the simulation
change total length of simulation

In this tutorial you will perform your first dynamical simulation of a mechanism. Chrono automatically computes all physical effects to show what would happen to the system in real life, and displays the simulation in realtime.

1

First, create a mechanism similar to the one described in the previus tutorial (a double-pendulum with two springs).

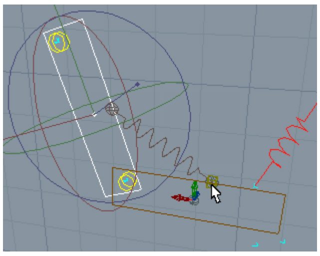


Figure 3.24

2

Press the ”play” button on the animation bar, at the bottom of the screen (it is the small ”¿” button). Well, now the simulation starts and you should see a realistic animation of the pendulum, oscillating up and down.

Figure 3.25

3

Now that the animation reached the last frame, you can use the menu ”Edit / Undo” to restore the animation to the original state.

Even if the animation has the ”auto rewind” option ON (see menu Windows / Animation settings), this does not imply that the physical simulation returns to the original state when time goes back to first frame. Hence you should use Undo to go back to the original state, otherwise when you start the animation the second time, the rigid bodies simply start moving from their last position -just like you simply continued the animation-.

4

While the animation runs, you can also see the time slider which moves on the time bar, showing the current frame.

5

If you press the ”play” button a second time, a third time, etc., without the Undo operation which restores the initial state, the simulation continues and obviusly the pendulum tends to stop, going to the rest position because of the damping effects. If so, you can use the IK tool to move it far from the static position, then press ”play” again and it will shake again, starting from the new position that you imposed with IK tool.

6

To change the length of the animation, enter a new amount of frames in the ”Total” filed of the animation bar, at the bottom of the screen.

Remember that each frame represents a 1/25th of second (in PAL mode) or 1/30th of second (in NTSC mode). If you want to show animation frames with longer or shorter time intervals, you can open the ”Animation settings” window (menu Windows / Animation settings...) and set ”custom” for the ”FPS” (Frames Per Second) field, then enter a new value.

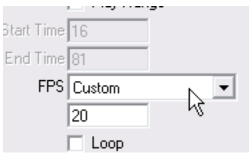


Figure 3.26

Each time you repeat the animation, the simulation is simply recomputed. The resulting motion is not stored, unless you turn on animation recording, as explained later in another tutorial.

At the bottom of the screen, on the animation bar, there is a long slider for time scrubbing. If you scrub forward/backward the time slider with the mouse, Chrono tries to perform the simulation between the growing time values (but it won’t ever simulate for backward scubbing, and won’t simulate if you drag the slider too fast, or you enter a different time in ”Current” field). Hence, to avoid troubles and misunderstandings, we suggest you to turn OFF the Chrono dynamical simulation when you want to scrub the time slider with the mouse (ex. go into the Property window of the System object, and turn OFF the simulation).

3.6 Tutorial: accuracy of dynamics

LEARN

system settings
integration step
tolerance

In this tutorial you can understand why and how to increase the precision of the physical simulator.

1

Create a triple pendulum, with three revolute links (the first revolute joins the upper rectangle and the ground reference).

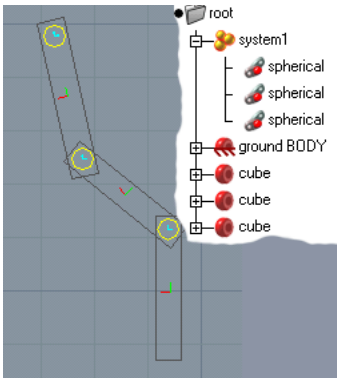


Figure 3.27

2

Press the ”play” button on the animation bar. Ok, the pendulum should swing as a chain linked to a roof.

3

If you have 25 frames per second and 150 frames in total, you saw about 6 seconds of animation. What happens if you want to proceed and see other seconds of simulation? Press ”play” again, to see other 6 seconds of simulation, then ress ”play” again, then again, then again, ...

4

After many seconds of simulation, you see that the pendulum will increase its speed, and the oscillation grows more and more. After other seconds, the system may ”explode” (the speeds rapidly reached infinite values and the pendulum parts disappeared).

As the system "explodes" because of high and fast oscillations, you start seeing also a warning on the status bar, at the bottom of the screen: "WARNING! Desired tolerance cannot be reached [..etc.]".

According to theoretical mechanics, this sudden increase in kinetic energy and the final explosion should not take place. So, how to explain this? The problem is that the physical simulation relies on a numerical integration method, which cannot reach infinite precision. Therefore, from frame to frame, some errors start to accumulate, and may cause the phenomenon of this example. If the errors accumulate in the worse way, some motions of the system can get unrealistic amplifications, which in their turn introduce even worse errors in next steps, and so on... in few steps, some rigid bodies will be shooten away very far from our sight: a divergence happened. Often the problem can be solved by enhancing the precision of the integration scheme.

5

Perform undo (or delete all and build another new pendulum). Now we will repeat the same experiment, but with higher precision in simulation settings.

6

In order to increase the precision of the integrator, open the property window of the System object. Change "Integration" from Euler to Runge-Kutta 4th order. Change also the time step from 0.04 to 0.02 because this is the time interval used by the simulation engine for the integration of the dynamical equations of motion (the smaller the step, the higher the precision).

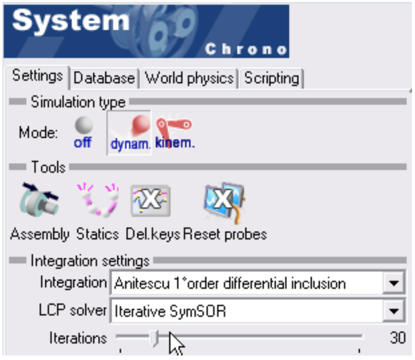


Figure 3.28

7

Press the play button to see 6 seconds of simulation, then press it again to see next 6 seconds, then again, then again, etc.. You can see that, this time, the simulation is much more precise, and the growth in kinetic energy does not happen.

The higher the precision, the slower the simulation. Depending on what you are simulating, you may do some tests to discover which is the best tradeoff between acceptable precision and good computation speed.

We suggest you to perform test simulations with the default settings (custom Eulero integrator, dt=0.04 seconds or so) because low precision can be acceptable for first tests, where you prefer fast playback. Later you can increase precision for more precise final simulations, where you can afford slower playback.

Systems without dissipation (ex: dampers, friction) like the pendulum of this example may diverge after a finite number of oscillation. But this number can be quite high if we use a short integration step and a precise integration method.

Keep in mind, however, that in real life there are no mechanisms without dissipation (even the most precise pendulum soon or later will stop oscillating because of the friction of the air or the friction in the joint). Nice to know, introducing some dissipation into the system (ex: some dampers) often make numerical divergence a lot less frequent if not impossible at all.

Also systems with very stiff springs, and affected by strong forces with noticeable nonlinearities may cause divergence if wrong simulation settings are used. Multibody specialists call such systems "stiff problems". It is not the situation of our pendulum, anyway.

If you meet a divergence, for the moment just stop the playback and use UNDO.

In some cases, if your system "explodes" and rigid bodies are shooten apart, maybe the cause is not numerical divergence (then the problem cannot be solved simply increasing the precision of the integrator). This is the case, for example, of the simulation of a mechanism which cannot be assembled, etc.

3.7 Tutorial: forces

LEARN

apply forces and torques
prismatic joint
change mass properties

In this tutorial you will apply a force to a system, you will change the mass of a body and the anchor point of a spring while the animation is running.

1

To begin, you should build two objects like the ones in figure (a small cube stacked on a large horizontal guide). Two simple analytic polyhedrons are enough for this test.

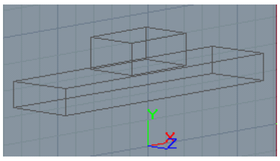


Figure 3.29

2

Now we are going to create a constraint which forces the upper cube to move along the other, like on a guide (the so called ”prismatic” joint). Therefore click on the Link tool and select ”prismatic” as Type.

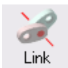


Figure 3.30

3

Create the constraint between the two objects like you did in the previous tutorials, but remember that the prismatic type is inserted with the translation axis perpendicular to the imput plane (so we suggest you to perform the link insertion by clicking on the left-side view, on the left input plane, like in figure).

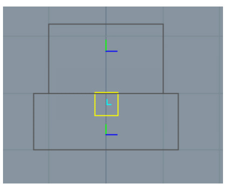


Figure 3.31

4

Do not press ”play” because both objects are free in space and would fall down. To avoid this, we must fix the horizontal cube by opening its property window (”Spec” tab) and turning ON the checkmark named ”Object locked to ground”.

5

Now you can test the motion of the upper cube with the IK tool from the front view (It should simply move horizontally respect to the other shape). This means that the prismatic joint works correctly.

6

Also add a spring which connects the moving cube to the ground, like in figure (see previous tutorial, on springs).

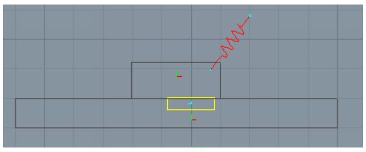


Figure 3.32

7

It is time to add a force: select the moving cube, click on the Force tool and set Strength=20 in tool preferences. To complete force creation, first click on the application point, then click a second time on the left, to indicate the direction of the force vector. An arrow should appear.

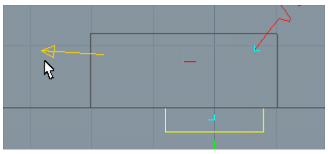


Figure 3.33

8

Press ”play” and see how the cube is pushed left because of the constant force.

9

Oscillations may last few seconds. Well, to make the things more interesting, do the following: set Loop=ON in Animation Settings window and press ”play” (the simulaition will last forever until you will stop it).

10

Now, while the animation is running, you can change parameters of the system: for example you can select the anchor point of the spring (the marker in the sublevel of the ”ground” object) and move it up and down, right and left by acting on his 3D handles. Note how you are really ”shaking” the moving cube, exploiting realtime interaction with the dynamical simulation.

11

Also try to change force modulus of the Force object (use its property window) or its direction.

The ability of performing runtime interaction with models is one of Chrono highlights. In this way, the user has an instant feedback on what happens if he changes some parameter.

Note, however, that leaving the property window opened while the simulation runs may slown down the playback speed.

3.8 Tutorial: masses, geometry

LEARN

add details to a rigid body
compute mass and inertia
change position of center of mass

In this tutorial you learn how to add geometric details to rigid bodies. Also, there are some hints on how to change by hand the position of the center of mass of a body, or how to compute it automatically, given the density of the material.

1

First of all, create a cube and link it to the ground using an prismatic joint, allowing horizontal translation only. (Remember that you must enter the position of the axis of the prismatic joint by clicking on the side view, as explained in the previous tutorial).

2

Create a spring which connects the cube to the ground, using the 'Spring' tool.

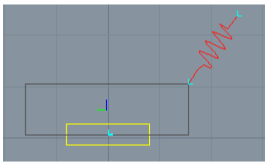


Figure 3.34

3

Select the "ground BODY" object in select window, and press the TAB key on the keyboard. This will make that object the *current object*, so that object which you wil create in future will be automatically placed into that sublevel. Also click on the "+" symbol on the left, to expand the level.

4

Create a polyhedron shaped like an "L". It will belong to the "ground" body, being placed in its sublevel.

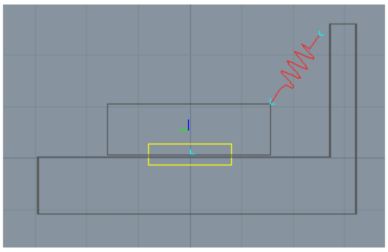


Figure 3.35

You can also drag and drop objects from and to rigid bodies. Also markers and force objects can be dragged and dropped from a rigid body to another.

You can use also cut and paste (see popup menu) instead of drag and drop, to move geometric objects between rigid bodies.

5

Now suppose we want the ”ground BODY” rigid body to be unlocked from the absolute reference, becoming moveable like the other object. Simply select it and open the property window, then uncheck the ”Object locked to ground” option.

6

Create a revolute joint between the ”L” shaped body (called ”ground BODY”, but now floating because just unlocked) and the absolute reference.

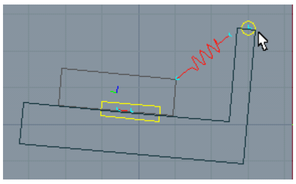


Figure 3.36

7

Test the degrees of freedom of this new system by picking the upper cube and moving it with the Chrono IK interactive kinematics tool (also the lower ”L” shape should swing up and down around the revolute joint)

8

Create a spring between the ”L” shape and the absolute reference, lke in these figures.

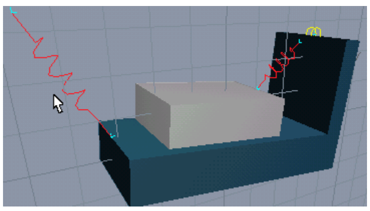


Figure 3.37

9

Press the animation play button and watch the oscillations.

10

Note that the position of handles is not correct for the ”L” shape. This is not a problem because you can move the handles in the proper place (the center of mass of the shape).

This task requires that you select the rigid body, then use the ”Move” tool (Transformation toolbar), set ”Target=Handles” and move handles with the mouse.

The position of center of mass (also called COG, from Center Of Gravity) is fundamental for exact dynamical simulations. In Chrono, the origin of handles is taken as center of mass, and X,Y,Z handle axis are taken as principal axes of inertia.

11

Note: a proper positioning of the center of mass cannot be always guessed by the user, so Chrono offers an automatic method which computes the correct postion of center of mass and also computes the total mass and moments of inertia of rigid bodies. Select the ”L” body and open its property window, then press the button named ”Recompute mass, COG, Inertia” in the Spec tab.

12

Look how the center of mass has been moved in the exact position, and see how the mass and inertia tensor has changed for both bodies.

The automatic computing of center of mass and total mass is obtained by means of a numerical quadrature (hence approximated, not exact). The space which contains the body is sampled many times, looking for points belonging to volumes contained in rigid body sublevel. If the rigid body is composed of thin structures (like thin rods, small bars, grillages, thin plates, etc.) it may happen that the position of center of mass (and total mass value) is affected by some rough approximation.

When performing automatic computation of mass and center of mass, avoid to use thin geometries, rods, spikes, small holes, thin shells, etc. It is better to compute mass using a basic approximation of the real shape, then add the little details later (this also makes the mass computation faster).

Avoid the use of volumes constructed by Nurbs if the surfaces are not perfectly closed. In fact the mass computation method assumes that all volumes are closed -otherwise, results may be wrong-.

The value of computed mass depends on material density. How can you set it? Simply go to rigid body property window and set the density value (in Kg/m^3). Default value is $1000Kg/m^3$, that is the density of water. Just as a reference, the density of steel is $7000Kg/m^3$, light alloys can be between $3000Kg/m^3$ and $5000Kg/m^3$.

13

Before pressing the animation play button again, remember that masses have been changed, and now the mechanism may be much heavier then before. If so, you may need to adjust and increase the stiffness of the springs.

3.9 Tutorial: engines

LEARN

create a motor
slider-crank mechanism
impossible or ill-posed mechanisms

This tutorial explains how to add engines to mechanisms. Also, you will see that motor settings can be easily customized. A slider-crank mechanism is used as example of motorized system.

1

Create a cylinder (representing a flywheel, used as crank) and a rectangular shape (representing the crank) as suggested in figure.

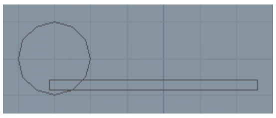
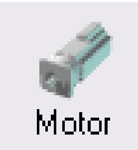


Figure 3.38

2

Click on the ”Motor” tool of the Chrono toolbar, this will allow you to add a link of type ”motor”. Note that this tool requires the same steps that you learned for all other link creations.



3

Use the ”Motor” tool to link the flywheel to the ground (pick flywheel, pick empty space, double click on center of flyweel). This done, the flywheel is constrained to rotate because of the motor.

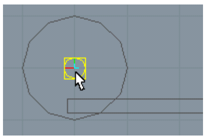


Figure 3.39

4

Use the ”Link” tool to create a link of type ”revolute” between the flywheel and the rod. (As joint position, click near the left side of the rod, where the rectangle is a bit overlayed to the wheel).

5

Use the ”Link” tool to create a link of type ”point-line” between the horizontal rod and the ground. (As line position, click near the right side of the rod).

Figure 3.40

6

Press the animation ”play” button and see what happens. The wheel should pull and push the rod on the horizontal line, reproducing the slider-crank behaviour.

7

Maybe you want the sliding direction to be aligned to wheel center: if so, you can expand the ”ground BODY” level in the select window, then select the proper marker and move or rotate it -since it is used as reference for the link, the sliding line will change accordingly-. This can be done even while the animation is running.

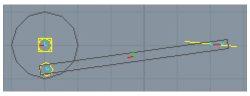


Figure 3.41

8

If you move that marker (which represent the reference of the ”point-line” link) too far in up or down direction, you end with a mechanism which cannot be assembled, or which cannot perform full 360 rotation. Such situation would cause a warning: if you press the ”Assembly” tool, the status bar shows the message ”WARNING: some constraints cannot be assembled [etc]”.

Sometimes it happens that valid mechanism are created at the beginning (i.e. they can be correctly assembled at t=0), but later, during their movement, they may be stretched because of motor rotations up to impossible configurations, and the ”WARNING..” message is then displayed. It is up to the user to avoid the design of such impossible mechanisms.

9

Maybe the motor rotates too slow for your tastes? If so, select the ”motor” link object in the select window, and open its property window. Go to the ”Spec” tab, then in the ”Custom” subtab.

10

Note that the motor is working in ”Impose speed mode”, so you are setting the instant angular speed of the two linked parts (i.e. the flyweel body, respect to the ground body). To cahnge the imposed speed, click on the button of the ”w(t)” function. A window now opens, showing the graph and the settings of the function.

11

In the field named ”C” you can enter an higher value, for example C=15 (this means that angular speed is constant with time, with constant value $\omega = 15[rad/s]$).

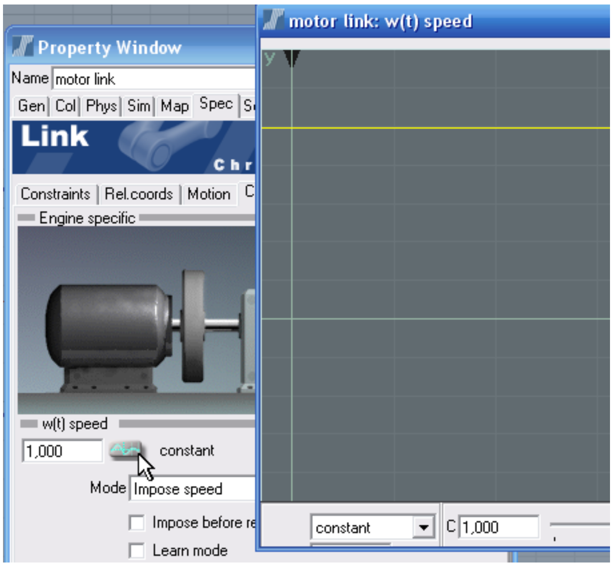


Figure 3.42

Most angular settings in Chrono are considered in radians. For example, 90 *degrees corresponds to* 1.57 *because* $1.57 = 90 \cdot (\pi/180)$. *Hence, also angular speed is by default expressed in* $[rad/s]$

12

Press the animation "play" button and see that the crank wheel rotates faster.

13

Try this: go back to the property window of the motor link, and open again the editor of the "w(t)" function. Change the function type to "sine". Settings for the sine function appear. Enter Frequency=0.4 and Amplitude=10.

13

Press animation play button and look how angular speed of the crank is oscillatory, because you are imposing an angular speed which is changing with sine function of time.

Lot of features of Chrono exploit this type of interactive function editors. These are called ChFunctions , and are documented in the . Thank to ChFunctions , it is easy to set non-linear functions of time (or space, etc.) for some properties of objects, such as in the case of the angular speed of this motor.

Maybe you are tempted to simulate this mechanism with extra-high angular speeds. However, despite a fast slider crank will rotate without problems in the true world, remember that you can meet numerical troubles with the simulation. In fact, if you simulate mechanisms which spin very fast, remember this rule of thumb: you need at least *10-20 integration steps for each turn. Hence, with default settings (step about 0.04 s), you can expect unacceptable results if some part is spinning faster than 2 revolutes per second (if so, you may need to set lower time steps for the integrator in "System" property window).*

3.10 Tutorial: chain tool

LEARN

create a chain
"placer" object
parametric linkages
four-bar mechanism

This tutorial explains how to use the Chrono Chain tool to create articulated mechanisms with minimum efforts. In fact, until now you learned to create mechanisms by entering rigid bodies and links with many tool operations; however you can use the Chain tool, which automatically creates many rod-like rigid bodies in sequence (also the needed links are created automatically), by simply entering the points where the joints must be created.

1

Be sure no object is selected.

2

Start the Chain tool, in Chrono toolbar (this is the tool which allows the creation of articulated sequences of rods).

3

Enter four points in the front view, from top to down. You will see that the Chain tool creates interactively the joints of the linkage, at each mouse click.

4

After the fourth click, use the popup menu (RMB on view) to accept the tool. The chain ends at the fourth entered point,and may look like the one depicted in figure.

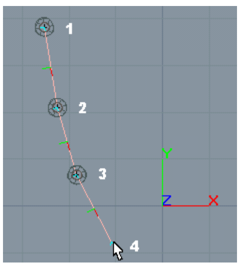


Figure 3.43

5

Now you can test the behavior of the linkage by selecting the last rod and using the IK tool, to drive it with the mouse. Also, you could press the animation play button and look this triple pendulum swinging.

6

Look at the select window: you will note that the Chain tool worked like a ”wizard” which quickly built a mechanism for you: there are rigid bodies, links, etc. Therefore, all usual operations on rigid bodies, markers, forces, etc, are still allowed (for example you can select two rigid bodies and create a spring between them, etc.)

7

In select window, expand the sublevel of the second rigid body (click on + symbol), select the ”placer” object, expand the sublevel of the placer -see figure-. Note that the ”placer” object contains a marker: the marker which represents the end of the rod, and which is used as reference by the connecting joint.

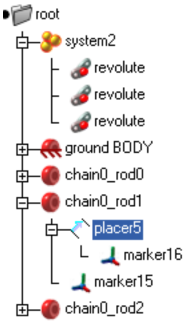


Figure 3.44

8

Select the ”placer” object: note that under the tool bar a ”position” slider appears. You can slide it with the mouse, and you will see that the length of the pink line which represents the placer object will change accordingly (also, the child marker will be moved, as it were anchored at the end of that line).

9

After you changed the length of one or two placer objects, click the Assembly tool, and the mechanism will be updated and assembled with the new rod lengths.

Figure 3.45

This means that the ”placer” object can be used to create ”parametric” mechanisms, where you can move or modify the placement of some details interactively, just by acting on sliders. If it is used to define the placement of markers, as in this example, it represents a good way to change the body-relative position of a joint.

In detail, the Chain tool automatically creates a placer object for each rod, so that the user has an easy way to change the length of each rod, by simply acting on the slider of the corresponding placer.

The Chain tool, by default, starts from the ground and ends free, but there are two buttons which allow you to make chains which start from ground and and to ground (closed kinematic loops) or start free and end to ground, or start free and end free.

How to make a chain which starts linked to an already-existing rigid body? Simply select the rigid body and use the chain tool: the chain will be attached to that body.

How to make a chain which starts linked to a rigid body and ends linked to another rigid body? Simply select both the start-body and the end-body, then use the chain tool: the chain will connect the two bodies.

10

Select all objects and delete them. We are going to demonstrate other features of the Chain tool.

11

Click on the Chain tool, and check the ”Ground end” button.

12

Select the ”motor” icon from the tool settings (this means that each entered point would be a motor, instead of the default revolute joint)

Figure 3.46

13

Click in the front view for the first point of the chain: you defined the position of the motor (between the ground and the first rod of the chain).

14

Select the ”revolute” icon from the tool settings, and enter the second, third and fourth point in 3D view, to create the three revolute joints of a four-bar mechanism.

15

Accept the tool after the last joint. You should get a four-bar mechanism like the one of the following figure. (Pay attention that the crank should be short, otherwise it won’t rotate all 360).

16

Press play, look the four-bar mechanism which rotates. If it rotates too slow, remember that you can select the motor link and modify the default speed, using the property window (Spec tab, Custom sub-tab).

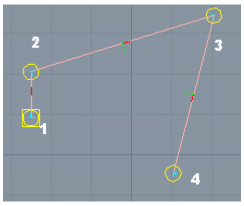


Figure 3.47

Later you can add details to the rough ”skeleton” which has been set up by the Chain tool (for example, you can drop some geometric objects into the rigid bodies, so that the rods can be rendered with nice shapes).

16

As an exercise, try to make a slider-crank mechanism like the one of the previous tutorial, but using the Chain tool. It should be much easier (it will take 3 mouse clicks, one for the motor, one for the crank revolute joint, one for the sliding line).

3.11 Tutorial: probes

LEARN

add ”probe” objects
measure variables
plot graphs
draw trajectories
draw 3D vectors

This tutorial explains how to use the Probe objects to record the variables of your simulations. You may need this feature if you want to study and analyze the graphs of joint reactions, or speeds of points, or distances, etc.

The Probe is an item without geometry (you can select it only in the select window: it does not appear in 3D views) and its task, in general, is to fetch variables from the Chrono object to which it is inserted. A probe object can work in different ways, for example it can create graphs of the recorded variable, or it can draw 3D trajectories in space, etc.

1

Create a four-bar mechanism like the one of the previous tutorial (be sure that the crank is motorized and capable of 360 of rotation).

2

Select the upper rod. Click on the Marker tool, which allows you to add a Marker object to the selected rigid body. Click on a point of the screen, to add the auxiliary marker (whose trajectory we want to draw).

3

You should end with a system like the one in the following figure. Note that the auxiliary marker belongs to the upper rigid body, hence it will move with that body.

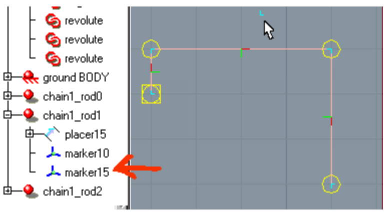


Figure 3.48

4

Use the select window to select the auxiliary marker, then click on the Probe tool, in Chrono toolbar. This will allow you to create a ”probe” object for the marker.

All Probe objects should be inserted as children of Chrono objects (links, rigid bodies, forces, markers, etc.). The Probe objects will monitor the variables of the parent object.

5

Before accepting the Probe tool, look at the tool settings: you can choose which variable (of the marker) has to be monitored by the probe. Also you can choose the type of action. Here, you must choose ”3D trajectory” as measure type. (Do not change the default variable ”context().p.pos”, that is the absolute position of coordinate system of the marker).

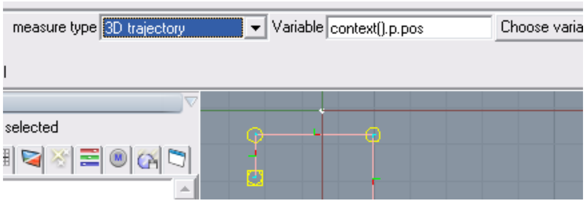


Figure 3.49

6

Accept the Probe tool. Note that a Probe object will be created in the sublevel of the marker object which you selected.

7

Open the property window of the Probe object, and go to the ”Spec” tab. Here, enter 60 as max number of points for 3D curve.

8

Press the animation play button. You will see that the auxiliary marker will leave a trailing trajectory plotted in 3D space, as in figure.

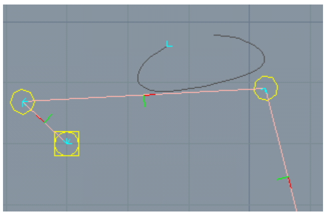


Figure 3.50

The plotted trajectory is a normal 3D spline, and you can find it in the System sublevel. Such line can be deleted, copied, modified, etc.

If you want to see the trajectory rendered in ray traced animations or pictures, select the line and change its ”invisible in photo realistic rendering” flag to OFF, in property window. Try also to change its thickness and rendering color.

9

Also, suppose we want to draw the speed vector of the auxiliary marker. Simply create another Probe for that marker (select the marker, click on the Probe tool).

10

This time, in the tool settings, we must choose measure type= ”3D vector plot”. Since we want to draw a vector, we also must specify which is the vector variable to monitor. Then, click on the ”Choose variable..” button and select ”p_dt.pos” from the tree of javascript variables of the marker (”p_dt” means the time derivative of the absolute coordinate of the marker, while ”pos” is the vector part of that coordinate -i.e. the speed-. See the Javascript).

When you clicked on the ”Choose variable..” button, the ”Variable browser” window opened. Expanding and clicking this tree of Javascript variables, the ”Variable” field is automatically filled for you -so you do not worry about reminding the name of variables-.

11

Accept the Probe tool. You should see that another probe object has been created into the marker sublevel.

12

Press the animation play button, and see how a three-dimensional arrow is drawn on the marker, representing its velocity vector. Such vector is modified in realtime, as the marker moves and changes its velocity.

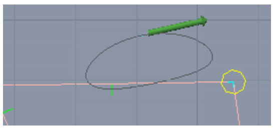


Figure 3.51

Maybe that arrow is too long (or too short), but this depends on how fast the parts are moving. If the excessive length clutters the view, or does not fit into the window, you can do this: open the property window of the probe, and set the ”Length scale” to a different value (ex: 0.1), to have the vector drawn in a lower scale. (There is a similar setting to scale also the thickness).

13

We will add a third probe, but this time it will monitor the reaction force in the rightmost revolute joint. To do so, select the link, click the Probe tool, set measure type = ”3D vector plot”, choose the variable ”react_force” (it is a vector) with the help of the ”Choose variable” browser, then accept the tool.

13

Press the animation play button, you will see that a vector is displayed in the revolute joint, showing the strength and the direction of the reacion exchanged between the rigid body and the ground.

14

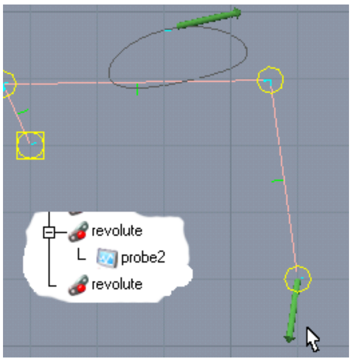


Figure 3.52

What if we want to record a graph showing the plot of a variable as a function of time? To explain this, we will add a fourth probe.
Select one of the links which represent the revolute joints. Click on the probe tool, press ”Choose variable”, choose the rel_angle variable and close the browser, then accept the tool. This probe will work in the default mode, that is ”Ch-function recorder”.

15

Press the animation play button. When the simulation has performed, select the last created probe and click on the ”Open plot” button which appears near the toolbar. You should see a window with the plotting of the rotation angle of the joint, as a recorded function of time.

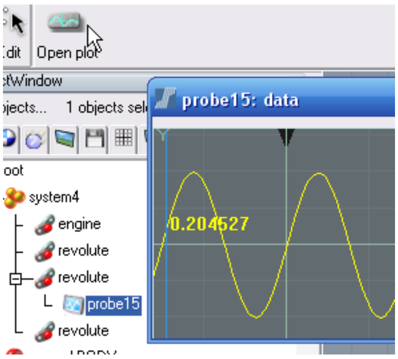


Figure 3.53

Use the right mouse button to open the pop-up menu for the plotting window: using ”File/import-export” you can export the recorded data in ascii format, for example if you want to use it in an Excel spreadsheet.

Use the pop-up menu ”File/Save as PostScript” to save the visible area of the graph into an .EPS file (Encapsulated PostScript). Such .eps figure can be imported in publishing software or in word processing programs (note: some programs will show the preview of the .eps figure as an empty box, but most laser printers will print it correctly).
Note: EPS files can be used also to make PDF documents, for example with Acrobat Distiller(TM).

Each Chrono object has lot of properties, and most of them can be accessed through Javascript variables. In fact the probe object uses the name of Javascript variables to fetch the properties to plot.
Note, however, that plotting 3D vectors or trajectories requires that you fetch a variable of type Vector (in the Variable browser, vectors have x,y,z sub variables, like the cases of react_force and p.pos, seen before).
Otherwise, when probes are in "Ch-function recorder" mode, you should fetch a scalar (mono-dimensional) variable, such as in the example of rel_angle.

3.12 Tutorial: linear actuators

LEARN

use linear actuators
Ch-Function of type 'sequence'
turn constraints on/off
easy handling of telescopic struts

Linear actuators can be used to impose distance between two parts. They act like hydraulic cylinders, which set the distance between the two spherical joints at the end of the telescopic strut. The user can set the value of the imposed distance as a ChFunction , function of time.

1

Create an articulated mechanism as the one of the following figure (it should resemble a backhoe loader, made of three free parts and a ground reference, connected by means of three revolute joints).

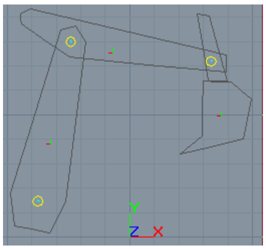


Figure 3.54

2

Be sure nothing is selected, then use the Actuator tool of the Chrono toolbar. Such tool works a bit like the Link and the Spring tools: it requires the selection of the two parts to connected, then it asks the position of the two constraint references.

3

For the 1st step of the Actuator tool, select the arm of the mechanism, on the left. As the 2nd step, click on the empty space, to mean that you want to constraint the arm to the ground. As 3rd step, click near the middle of the arm (this means the first end of the actuator). As 4rd step, click on the bottom-right: this will be the other end of the actuator. An actuator will be created: it could represent an hydraulic actuator between the ground and the arm.

3

Use again the Actuator tool: this time you will create an actuator between arm and forearm. For the 1st step of the Actuator tool, select the arm of the mechanism, on the left. As the 2nd step, click on forearm. As 3rd step, click near the left of the arm (this means the first end of the actuator). As 4rd step, click on the left side of the forearm: this will be the other end of the actuator. The actuator will be created.

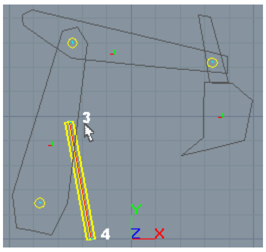


Figure 3.55

4

Use again, for the last time, the Actuator tool. Do the four step to create a linear actuator between the forearm and the spoon. The final result should be similar to this figure.

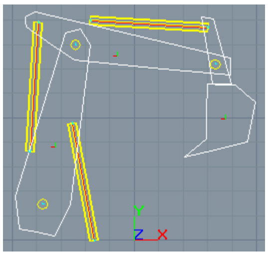


Figure 3.56

5

Press the animation play button: you should see that the mechanism remains steady (it does not fall) because the three cylinders support the weight, and their lenght is constrained.

6

To change the imposed distance, open the property window of the last linear actuator link. Go to the Spec tab, then in the Custom tab. Here you can find the ”d(t) extension” ChFunction . You can use this function to impose a time-varying length.

7

For example: edit the d(t) function, opening its settings window and changing it from ’recorder’ to ’SEQUENCE’ mode (the ’sequence’ settings will be shown). Switch the type of the function to insert (near the ’Add function’ button), as ”Const.acc” type, and press the ’Add function’ to insert it. A smooth ramp will be added. Open the settings of this sub function,by accessing the ’segment function’ editing window, on the bottom (for this ramp, set its h value as h=0.1, then close its window).

Add another sub function to the sequence: this time add a function of ’sine’ type. Check the ”C0” flag of the sequence editor, so that the sine sub-function will have C0 continuity with the previous (the smooth ramp). Also set ”duration” to 6 seconds, at least. Then, open the setting of this sinusoidal segment, to set its frequency to 0.5 and amplitude to 0.02. The final result whould be an initial ramp followed by a slight sinusoidal function. Close all windows.

8

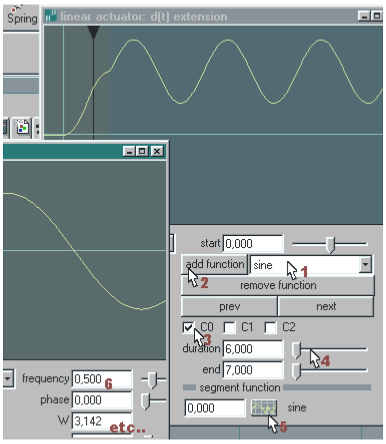


Figure 3.57

Press the animation play button: this time the lenght of the linear actuator is not constant because it follows the d(t) function which you just modified. Therefore, the mechanism will oscillate.

To record the force required to the actuator for this motion, select it and use the Probe tool to create a probe object which monitors the link variable called "react_force.x".

9

Try to move the mechanism with the Chrono IK tool: it is impossible. This is not an error, it is exactly what it is required by the three linear actuators, since they are imposing a distance between their ends, and this causes the stiffness.

10

Therefore, if you want to move the mechanism by hand (with the Chrono IK tool) you must set the link to "inactive" or "learning" state. Do the following: select the three 'linear actuator' links at once, then open the property window, go to the Spec tab, go to the Custom tab, set the "learning mode" checkmark to ON. You should see that the wireframes of the tree actuators turn green (this means that their constraints are free). Now you can use the Chrono IK tool without problems: the mechanism will follow.

11

To improve the look of the linear actuators, you can use the following trick. Load the objects chrono/parts/mech_Actuator_outer(1).r3d and chrono/parts/mech_Actuator_inner(2).r3d, for example using the popup menu 'paste from file'. Then, cut and paste them (in the proper order) into the 'linear actuator' sublevel. Repeat this for the other two linear actuators. Press the Assembly tool: note how the loaded cylinders will be updated to match the direction of the parent actuator, hence creating 'telescopic struts'.

12

To show a cleaner wireframe, now you can switch on the default yellow representation of the actuators (the telescopic cylinders are enough), so go to the Wire tab of the property window of the three actuators, and select "Hide geometry".

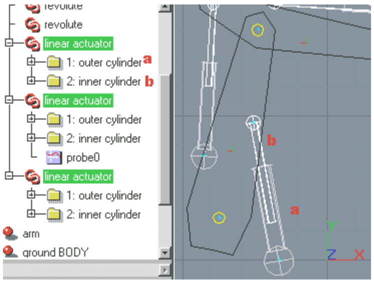


Figure 3.58

Why using the "Hide geometry" and not the simple "Invisible in realtime rendering"? Because the latter would also hide the two cylinders contained in the link sublevel, while the former only hides the yellow default wireframe, but still shows the children objects.

This trick of putting two cylinders in the sublevel of Actuator links to simulate telescopic struts is only for aesthetical purposes! In fact the mass of the two telescopic cylinder is NOT taken into account (if you need such effect, you should create a telescopic strut as an assembly of two rigid bodies with a prismatic joint and a linear actuator between them, and two spherical joints at the ends, but this often is not necessary -hence the trick discussed above-)

How does this trick work? The rule is this. Actuator links have a feature which automatically align the Y axis of the first TWO children sub-objects to the main direction of the actuator, while keeping the origin of the first subobject at the first actuator end, and the second at the second actuator end. In fact, you can create your own telescopic struts to be used as subobjects (be sure that their handles are aligned to show the Y axes as main direction, and the handle origin must coincide with the end, where there should be the spherical joint). Also, you can modify the ones which are providen as example on disk (but do not resize or skew or stretch the handles: if you need to resize their default value, please act directly on the object contained in their sublevels, namely the sphere and the cylinder).

3.13 Tutorial: Create keyframe curves

LEARN

record keyframes in choreographs
switch ON/OFF the simulation engine
hand-edit keyframes

Until now, you learned that each time you press the animation ’play’ button, Chrono simulation is computed on the fly. In fact, if you play the simulation, you would have only one way to repeat it exactly a second time after rewinding: you should execute an Undo, then simulate it again.

Otherwise, maybe you may to record the result of a simulation in a keyframe animation, so that you can scrub the time slider back and forth, etc, as you do for usual animations based on R3D choreography and keyframe curves. If needed, this can be accomplished by recording the result of a Chrono simulation into keyframes, as suggested in this tutorial.

1

Create a swinging pendulum, like the one of the following figure, for example by using the Chain tool. (This is for test purposes only: whatever mechanism would fit).

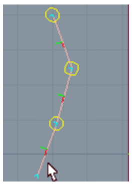


Figure 3.59

2

Shift-select all the rigid bodies, and open the property window. Note that all rigid bodies have the ’Animation protected’ setting ON by default: you must turn it OFF (so that Realsoft3D can automatically create keyuframes for them when it detects their motion).

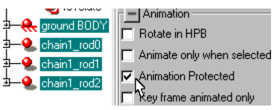


Figure 3.60

3

Unselect all bodies. Now you can turn ON the animation recording (on the animation bar, on the bottom of the screen), to enable automatic keyframing.

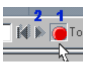


Figure 3.61

4

Press the animation play button (near the animation recording button) and wait for tha simulation to reach the end of the time bar.

The simulation may run a bit slower than in the case of simple simulation without creation of keyframing. In fact, the creation of keyframes (i.e. with the red 'recording' button tured ON) can slow down playback, expecially if you have lot of objects in your scene, because many curve interpolations must be created.

5

Important: after the animation reached the end of time, you must turn OFF the simulation engine of Chrono, because from now on, you can simply use the recorded keyframes. Therefore, do the following: open the property window for the System object, go to the Spec tab, and set 'Simulation type - Mode' to 'OFF, no analysis'.

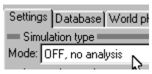


Figure 3.62

6

Once you performed the previous steps, you are ready to watch the simulation as recorded in keyframes. For example, you can also scrub the time slider back and forth: there are no problems because the motion of the pendulum isn't simulated anymore, it is stored in choreographs. Try to move the time slider, or press play button many times: the animation is always the same (since it was recorded).

7

If you want to understand what happened, you can open the choreography window. Use the menu Animation/choreography, and you will see the recorded translations and rotations for all the rigid bodies. (The list of the keyframed objects can be seen on the left of the choreography window, as in the following figure. Double-click the object to show its recorded variables).

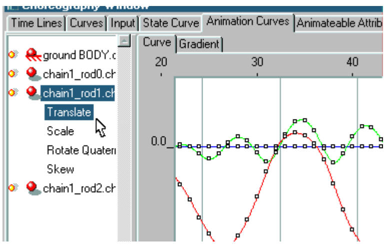


Figure 3.63

Now you can also hand-edit the animation curves by deleting or moving some points. This can be useful if you want to cut away undesired vibrations, motions, etc.

8

Last hint: suppose you want to repeat the recording because you forgot something: now all rigid bodies are full of keyframes, so you should delete all the choreographs, one by one, in the choreography window (a pretty tedious work). However there is a faster way: go to the property window of the System object, go to the Spec/Database tab, and press the 'Delete keys' icon: it will delete all the recorded motions of the rigid bodies (then you are ready to turn ON again the simulation engine and repeat all the tutorial).

You can see if rigid bodies (and other Realsoft3D objects) have some recorded choreographs because their label is shown in bold typeface in the select window.

While the System object is in 'OFF, no analysis' mode, no physical computations are performed at all, so (for example) you cannot expect that Probe objects are updated during a simple playback based on recorded keyframes, neither javascript programs in Control objects are executed, etc.

Also markers and links have the 'Animation protected' flag turned ON by default, and you seldom need to turn it OFF because you do not need to keyframe them.

Once you recorded the simulation into keyframes, you could also delete all the Link objects (and the System object too: in fact the simulation engine would be inactive anyway!) because from now on, you just rely on choreographs for the motion of the rigid bodies (however, this is NOT a wise choiche, because maybe that later you need the constraints again, to repeat the physical simulation with different settings).

3.14