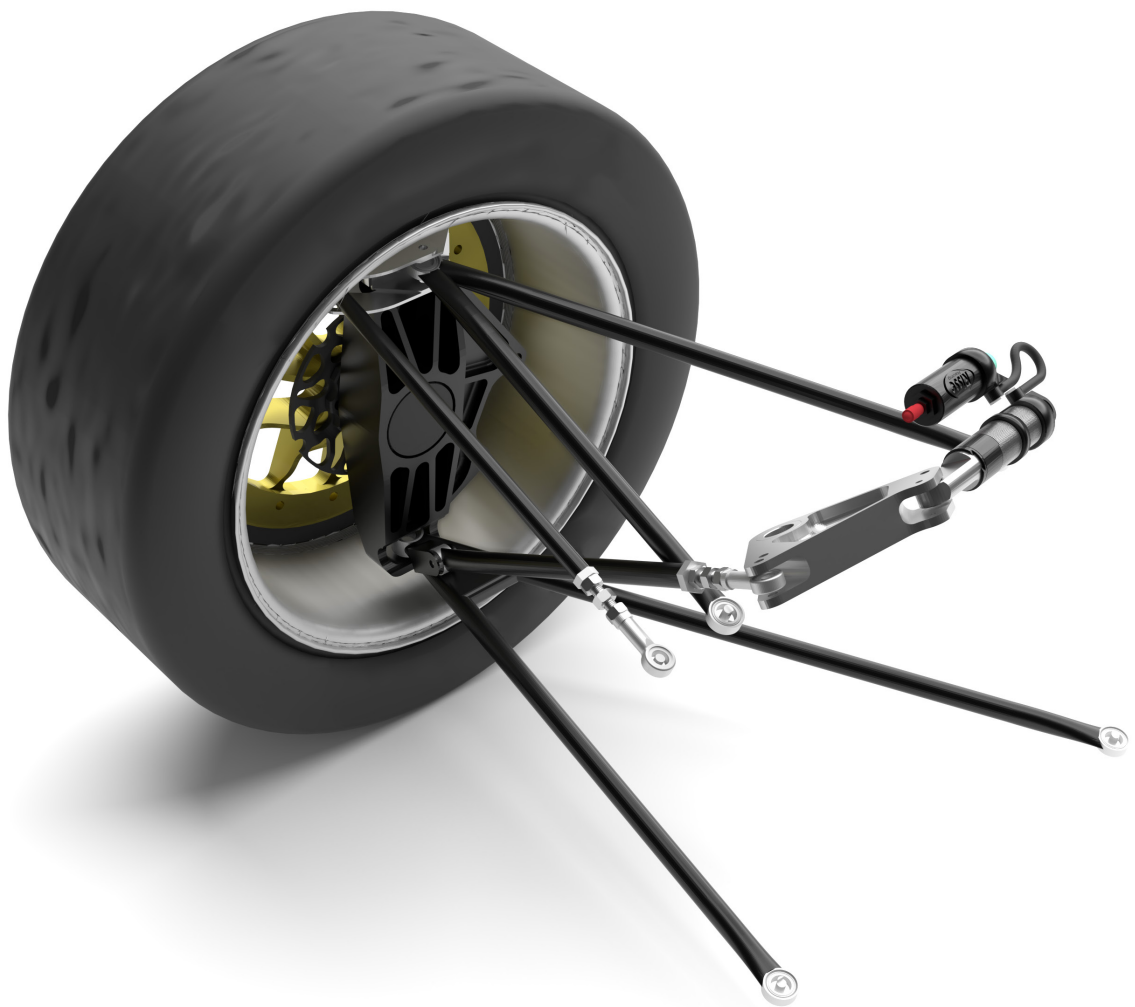


# Motion Simulation and Mechanism Design

with SOLIDWORKS® Motion 2019



Kuang-Hua Chang Ph.D.

Visit the following websites to learn more about this book:

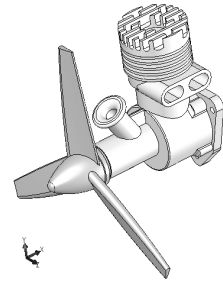


[amazon.com](https://www.amazon.com)

[Google books](https://books.google.com)

[BARNES & NOBLE](https://www.barnesandnoble.com)

## Lesson 1: Introduction to *SOLIDWORKS Motion*



### 1.1 Overview of the Lesson

This lesson intends to provide you with a brief overview on *SOLIDWORKS Motion*. *SOLIDWORKS Motion*, formerly called *COSMOSMotion* (*SOLIDWORKS* 2008 and before), is a virtual prototyping tool that supports animation, analysis and design of mechanisms. Instead of building and testing physical prototypes of the mechanism, you may use *SOLIDWORKS Motion* (also called *Motion* in this book) to evaluate and refine the mechanism before finalizing the design and entering the functional prototyping stage. *Motion* will help you analyze and eventually design better engineering products that involve moving parts. More specifically, the software enables you to size motors and actuators, determine power consumption, layout linkages, develop cams, understand gear drives, size springs and dampers, and determine how contacting parts behave, which would usually require tests of physical prototypes. With such information, you will gain insight on how the mechanism works and why it behaves in certain ways. You will be able to modify the design and often achieve better design alternatives by using the more convenient and less expensive virtual prototypes. In the long run, using virtual prototyping tools, such as *Motion*, will help you become a more experienced and competent design engineer by gaining insights on the mechanics of parts interaction in mechanisms.

In this lesson, we will start with a brief introduction to *Motion* and discuss various types of physical problems that *Motion* is capable of solving. We will then discuss capabilities offered by *Motion* for creating motion models, conducting motion analyses, and viewing motion analysis results. In the final section, we will preview examples employed in this book and topics to learn from these examples.

Note that materials presented in this lesson will be kept brief. More details on various aspects of the mechanism design and analysis using *Motion* will be given in later lessons.

### 1.2 What is *SOLIDWORKS Motion*?

*SOLIDWORKS Motion* is a computer software tool that supports engineers to analyze and design mechanisms. It is a module of the *SOLIDWORKS* product family developed by *Dassault Systèmes SOLIDWORKS Corporation*. This software supports users to create virtual mechanisms that answer general questions in product design as described next. A single piston engine shown in Figures 1-1 and 1-2 will be used to illustrate some typical questions, such as the following:

1. Will the components of the mechanism collide or interfere in operation? For example, will the connecting rod collide with the inner surface of the piston or the inner surface of the engine case during operation?
2. Will the components in the mechanism you are designing move according to your intent? For example, will the piston stay entirely in the piston sleeve? Will the system lock up when the firing force aligns vertically with the connecting rod?

3. How much torque or force does it take to drive the mechanism? For example, what will be the minimum firing load to move the piston? Note that in this case, proper friction forces must be added to simulate the resistance between parts in contact in the mechanism before a realistic firing force can be calculated.
4. How fast will the components move; e.g., the linear velocity of the piston?
5. What is the reaction force or torque generated at a connection (also called *joint* or *constraint*) between components (or bodies) during motion? For example, what is the reaction force at the joint between the connecting rod and the piston pin? This reaction force is critical since the structural integrity of the piston pin and the connecting rod must be ensured; i.e., they must be strong and durable enough to sustain the firing load in operation.

The modeling and analysis capabilities in *Motion* will help you answer these common questions accurately and realistically, as long as the motion model is properly defined.

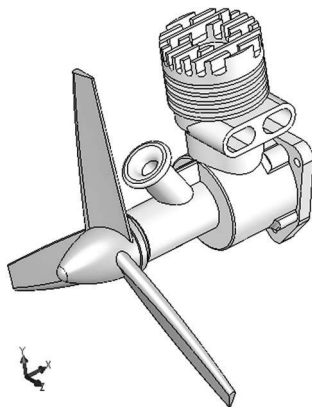


Figure 1-1 A Single Piston Engine (Unexplode View)

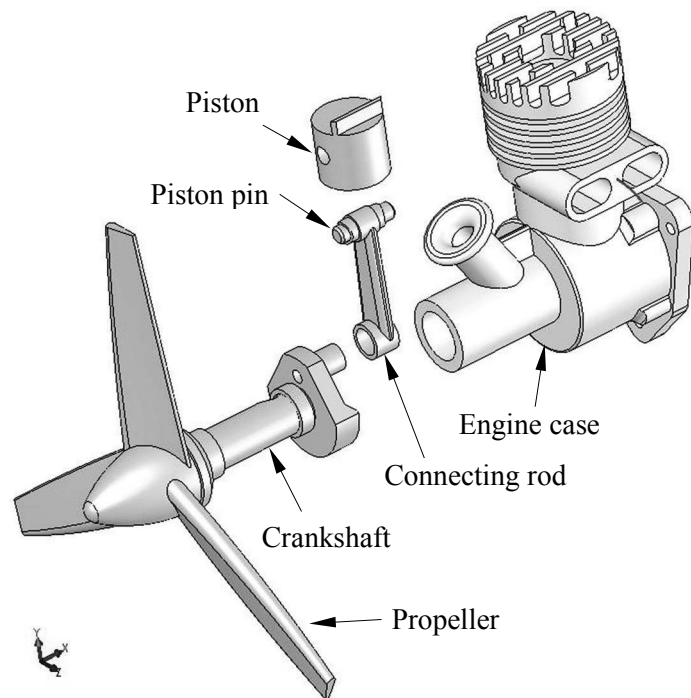


Figure 1-2 Single Piston Engine (Explode View)

The capabilities available in *Motion* also help you search for better design alternatives. A better design alternative is very much problem dependent. It is critical that a design objective be clearly defined by the designer up front before searching for better design alternatives. For the engine example, a better design alternative could be a design that reveals:

1. A smaller reaction force applied to the connecting rod, and
2. No collisions or interference between components.

One of the common approaches for searching for design alternatives is to vary the component sizes of the mechanism. In order to vary component sizes for exploring better design alternatives, the parts and assembly must be adequately parameterized to capture design intents. At the parts level, design parameterization implies creating solid features and relating dimensions so that when a dimension value is changed, the part can be rebuilt properly. At the assembly level, design parameterization involves defining assembly mates and relating dimensions across parts. When an assembly is fully parameterized,

a change in dimension value can be propagated automatically to all parts affected. Parts affected must be rebuilt successfully, and at the same time they will have to maintain proper position and orientation with respect to one another without violating any assembly mates or revealing part penetration or excessive gaps. For example, in this single-piston engine assembly, a change in the bore diameter of the engine case will alter not only the geometry of the engine case, but also all other parts affected, such as the piston, piston sleeve, and even the crankshaft, as illustrated in Figure 1-3. Moreover, they all have to be rebuilt properly and the entire assembly must stay intact through assembly mates.

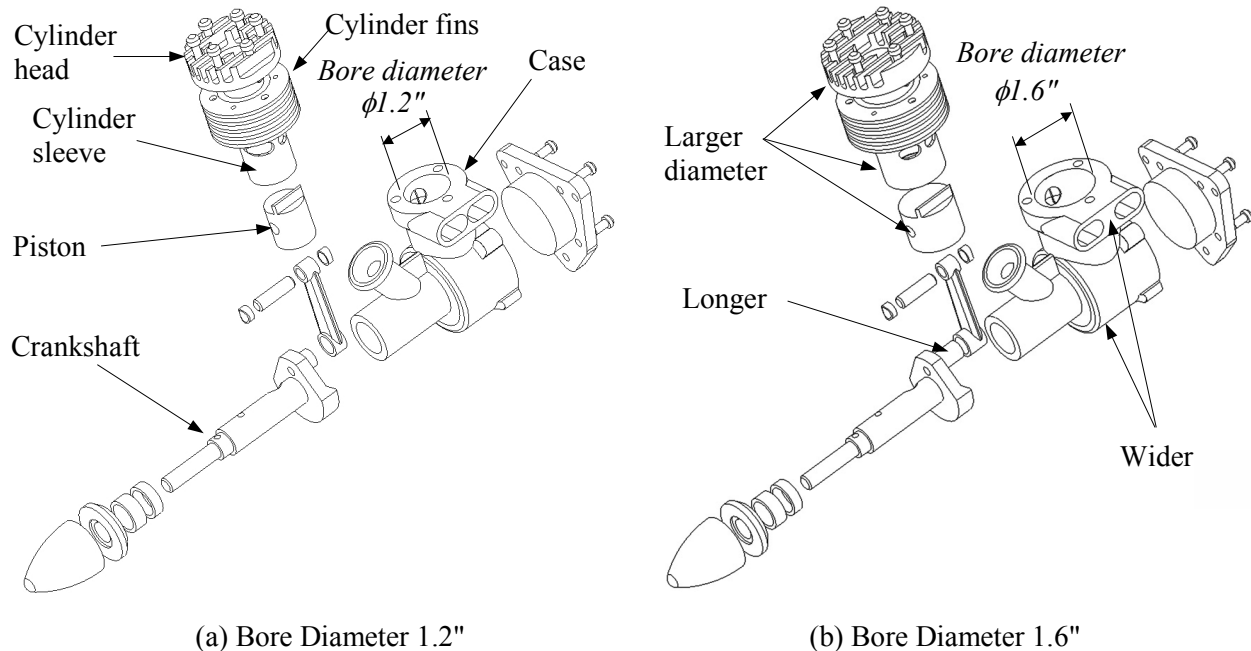


Figure 1-3 A Single-Piston Engine—Explode View

### 1.3 Mechanism Design and Motion Analysis

A mechanism is a mechanical device that transfers motion and/or force from a source to an output. It can be an abstraction (simplified model) of a mechanical system represented as a linkage. A linkage consists of links (also called bodies), which are connected by joints (or connections), such as a revolute joint, to form open or closed chains (or loops, see Figure 1-4). Such kinematic chains, with at least one link fixed, become mechanisms. In this book, all links are assumed rigid. In general, a mechanism can be represented by its corresponding schematic drawing for discussion and presentation purposes. For example, a slider-crank mechanism characterizes the engine motion, as shown in Figure 1-5, which is a closed loop mechanism.

In general, there are two basic types of motion problems that you will have to solve in order to answer questions regarding mechanism analysis and design: kinematic and dynamic.

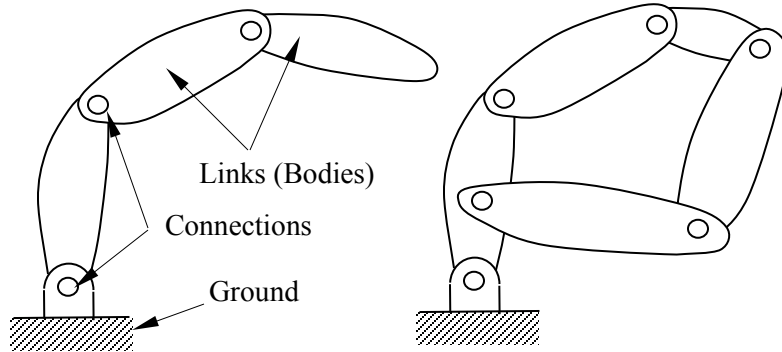
Kinematics is the study of motion without regard for the forces that cause the motion. A kinematic mechanism can be driven by a servomotor so that the position, velocity, and acceleration of each link of the mechanism can be analyzed at any given time. Typically, a kinematic analysis is conducted before the dynamic behavior of the mechanism can be simulated properly.

Dynamic analysis is the study of motion in response to externally applied loads. The dynamic behavior of a mechanism is governed by Newton's laws of motion. The simplest dynamic problem is the

particle dynamics introduced in sophomore Dynamics class—for example, a spring-mass-damper system shown in Figure 1-6. In this case, motion of the mass is governed by the following equation derived from Newton's second law,

$$\sum F = p(t) - kx - c\dot{x} = m\ddot{x} \quad (1.1)$$

where  $(\cdot)$  appearing on top of the physical quantities represents time derivative of the quantities,  $m$  is the total mass of the block,  $k$  is the spring constant, and  $c$  is the damping coefficient.



(a) Open Loop Mechanism      (b) Closed Loop Mechanism

Figure 1-4 General Mechanisms

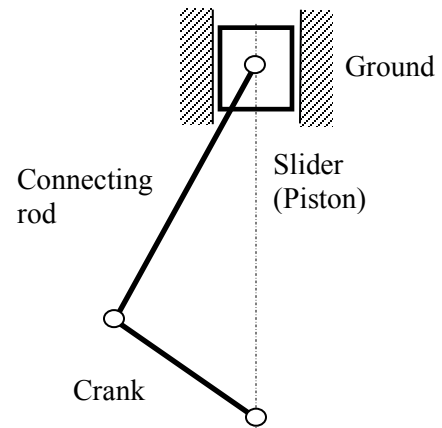


Figure 1-5 Schematic View of the Engine Motion Model

For a rigid body, mass properties (such as the total mass, center of mass, moment of inertia, etc.) are taken into account for dynamic analysis. For example, motion of a pendulum shown in Figure 1-7 is governed by the following equation of motion,

$$\sum M = -mg \sin\theta = I\ddot{\theta} \quad (1.2)$$

where  $M$  is the external moment (or torque),  $I$  is the mass moment of inertia of the pendulum ( $I = m\ell^2$ ),  $m$  is the pendulum mass,  $g$  is the gravitational acceleration, and  $\ddot{\theta}$  is the angular acceleration of the pendulum.

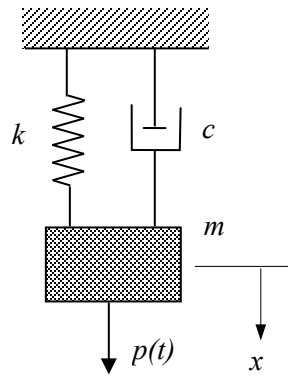


Figure 1-6 The Spring-Mass-Damper System

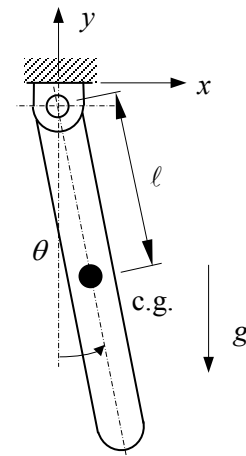


Figure 1-7 A Simple Pendulum

Dynamic analysis of a rigid body system, such as the single piston engine shown in Figure 1-3, is a lot more complicated than the single body problems. Usually, a system of differential and algebraic equations governs the motion and the dynamic behavior of the system. Newton's law must be obeyed by individual bodies in the system at all times. The motion of the system will be determined by the loads acting on the bodies or joint axes (e.g., a torque driving the system). Reaction forces and moments at the joints hold the bodies together.

Note that in *Motion*, you may create a kinematic analysis model; e.g., using a motion driver (linear or rotary motor) to drive the mechanism; and then carry out dynamic analyses. In dynamic analysis, the inertia of the bodies will be taken into account for analysis; therefore, reaction forces will be calculated between bodies.

#### 1.4 *SOLIDWORKS Motion* Capabilities

##### *Overall Process*

The overall process of using *SOLIDWORKS Motion* for analyzing a mechanism consists of three main steps: model generation, analysis (or simulation), and result visualization (or post-processing), as illustrated in Figure 1-8. Key entities that constitute a motion model include servomotors that drive the mechanism for kinematic analysis, external loads (force and torque), force entities such as spring and damper, and the initial conditions of the mechanism. Most importantly, assembly mates must be properly defined for the mechanism so that the motion model captures essential characteristics and closely resembles the behavior of the physical mechanism.

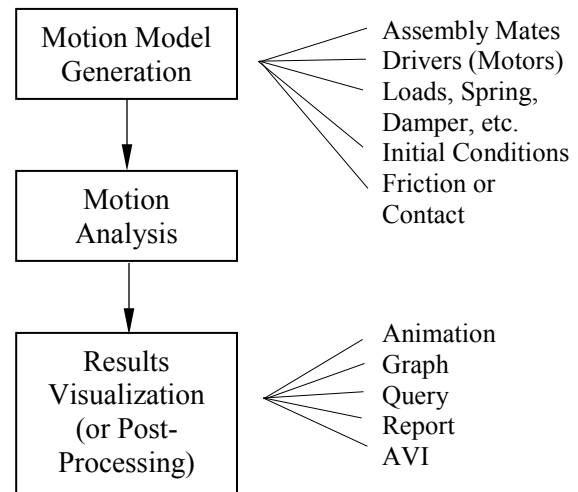


Figure 1-8 General Process of Using *SOLIDWORKS Motion*

The analysis or simulation capabilities in *Motion* employ simulation engine *ADAMS/Solver*, which solves the equations of motion for the mechanism. *ADAMS/Solver* calculates the position, velocity, acceleration, and reaction forces acting on each moving part in the mechanism. Typical simulation problems, including static (equilibrium configuration) and motion (kinematic and dynamic) are supported. More details about the analysis capabilities in *Motion* will be discussed later in this lesson.

The analysis results can be visualized in various forms. You may animate motion of the mechanism or generate graphs for more specific information, such as the reaction force of a joint in time domain. You may also query results at specific locations for a given time. Furthermore, you may ask for a report on results that you specified, such as the acceleration of a moving part in the time domain. You may also save the motion animation to an AVI for faster viewing and/or showing motion animation on other computers.

##### *Operation Mode*

*Motion* is embedded in *SOLIDWORKS*. It is indeed an add-in module of *SOLIDWORKS*. Transition from *SOLIDWORKS* to *Motion* is seamless. All the solid parts, materials, assembly mates, etc. defined in *SOLIDWORKS* are automatically carried over into *Motion*. *Motion* can be accessed through menus and windows inside *SOLIDWORKS*. The same assembly created in *SOLIDWORKS* is directly employed for creating motion models in *Motion*. In addition, part geometry is essential for mass property computations in motion analysis. In *Motion*, all mass properties calculated in *SOLIDWORKS* are ready for use. In addition, the detailed part geometry supports interference check (as well as detects contact between bodies) for the mechanism during motion simulation in *Motion*.

### User Interfaces—*MotionManager* Window

User interface of *Motion* is embedded in *SOLIDWORKS*. *SOLIDWORKS* users should find that it is straightforward to maneuver in *Motion*. The main interface for using *Motion* is through the *MotionManager* window, as shown in Figure 1-9. The *MotionManager* is a separate window that is used to create and play animations as well as conduct motion analysis. When you open an existing assembly (or part) in *SOLIDWORKS*, the *Motion Study* tab (with a default name *Motion Study 1*) will appear at the bottom of the graphics area. Clicking the *Motion Study* tab will bring up the *MotionManager* window. As shown in Figure 1-9, the user interface window of *MotionManager* consists of *MotionManager* tree (or *Motion* browser), *Motion* toolbar, filters, timeline area, etc.

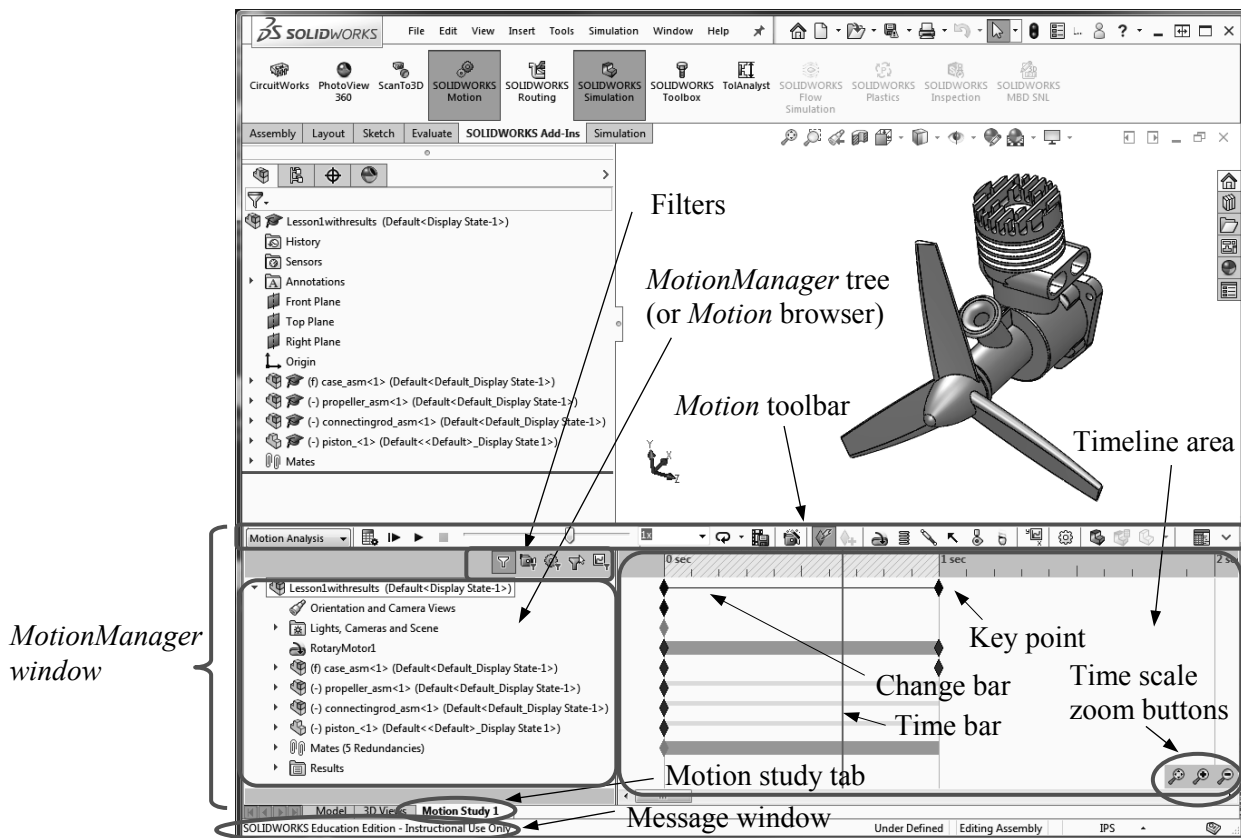


Figure 1-9 User Interface of *SOLIDWORKS Motion*—*MotionManager* Window

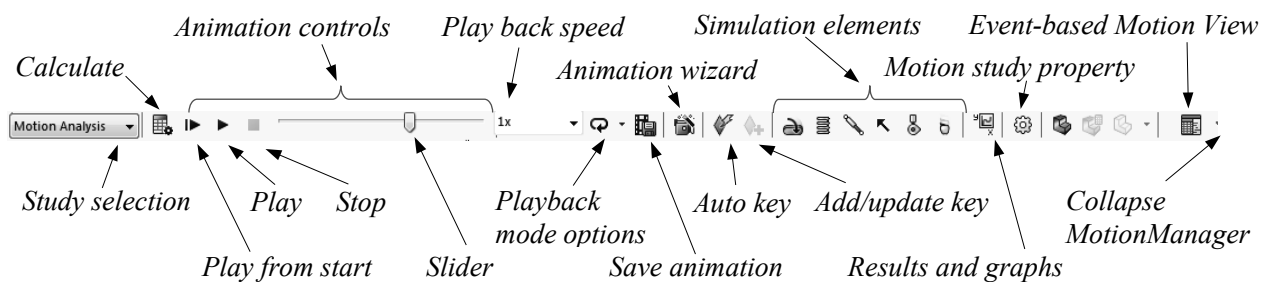
### *MotionManager* Tree (or *Motion* Browser)

Components are mapped from *SOLIDWORKS* assembly into *MotionManager* tree automatically, including root assembly, parts, sub-assemblies, and mates. An *Orientation and Camera Views* node is also added. This node will help you create animation in a specific view as desired, but has less to do with the motion simulation. Each part and sub-assembly can be expanded to show its composing components. Motion entities, such as spring and force, will be added to the tree once they are created. A result node will be added to the tree once a motion analysis is completed and result graphs are created. Similar to *SOLIDWORKS*, right clicking a node in the *MotionManager* tree will bring up command options that you can choose to modify or adjust the *Motion* entities.



Motion Toolbar

The *Motion* toolbar shown in Figure 1-9 (and more detailed in Figure 1-10) provides major functions required to create and modify motion models, create and run analyses, and visualize results. The toolbar includes a type of study selection (for choosing from animation, basic motion, or motion analysis), calculate, animation controls, playback speed and mode options, save options, animation wizard, key point controls, simulation elements, results and graphs, and motion study property. As you move the mouse pointer over a button, a brief description about the functionality of the button appears. For example, if you move the mouse pointer close to the *Spring* button, a brief description about this button will appear next to it as well as in the *Message* window, located at the bottom left corner of the *MotionManager* window, as shown in Figure 1-9. You may want to click some of the buttons and become more familiar with their functions. The buttons in *Motion* toolbar and their functions are summarized in Table 1-1 with more details.

Figure 1-10 The *Motion* Toolbar

Note that you may choose *Animation*, *Basic Motion*, or *Motion Analysis* for study. You can use *Animation* to animate simple operation of assemblies, such as rotation, zoom in/out, explode/collapse assembly, etc. You may also add motors to animate simple kinematic motion of the assembly.

*Basic Motion* supports you for approximating the effects of motors, springs, collisions, and gravity on assemblies. *Basic Motion* takes mass into account in calculating motion. The computation is relatively fast, so you can use this for creating presentation-worthy animations using physics-based simulations.

*Motion Analysis* (only available when *SOLIDWORKS Motion* is added in) provides you accurate simulation and analysis on the effects of motion entities (including forces, springs, dampers, and friction) on an assembly. *Motion Analysis* uses computationally strong kinematic solvers, and accounts for material properties as well as mass and inertia in the computations. You can also use *Motion Analysis* to plot simulation results for further analysis.

As a general rule of thumb, you may use *Animation* to create presentation-worthy animations for motion that does not require accounting for mass or gravity; use *Basic Motion* to create presentation-worthy approximate simulations of motion that account for mass, collisions, or gravity; and use *Motion Analysis* to run computationally strong simulations that take the physics of the assembly motion into consideration. *Motion Analysis* is the most computationally intensive of the three options.

Both *Animation* and *Basic Motion* are available in core *SOLIDWORKS*. *Motion Analysis* is only available with the *SOLIDWORKS Motion* add-in to *SOLIDWORKS*. Before using this book, you are encouraged to check with your system administrator to make sure the *Motion* module has been installed to your computer.
























Filters

Filters can be used to limit the *MotionManager* tree by including only animated, driving, selected or simulation results. Filters help make a cleaner display of entities in the *MotionManager* tree.

Timeline Area

The area to the right of the *MotionManager* tree is the timeline area, which is the temporal interface for animation. The timeline area displays the times and types of animation events in the motion study.

Table 1-1 The Function Buttons in *Motion* Toolbar

Symbol	Name	Function
	Calculate	Calculates the current simulation. If you alter the simulation, you must recalculate it before replaying.
	Play from start	Reset components and play the simulation. Use after simulation has been calculated.
	Play	Play the simulation beginning at the current timebar location.
	Stop	Stop the animation.
	Playback mode: Normal	Play from beginning to end once.
	Playback mode: Loop	Continuously play, from beginning to end, then loop to beginning and continue playing.
	Playback mode: Reciprocal	Continuously play, from beginning to end, then reverse—play from end to beginning.
	Save animation	Save the animation as an AVI movie file.
	Animation wizard	Allow to quickly create rotate model, explode or collapse animation.
	Auto key	Click to automatically place a new key when you move or change components. Click again to toggle this option.
	Add/update key	Click to add a new key or update properties of an existing key.
	Motor	Create a motor for motion analysis.
	Spring	Add a spring between two components.
	Damper	Add a damper between two components.
	Force	Create a force for motion analysis.
	Contact	Create a 3D contact between selected components.
	Gravity	Add gravity to the motion study.
	Results and graphs	Create results and graphs.
	Motion study properties	Define motion study solution parameters.
	Simulation setup	Set up stress analysis on one or more components.
	Event-based motion view	Switch from the timeline view to an event-based view.
	Timeline view	Switch from event-based view to a timeline view.
	Collapse <i>MotionManager</i>	Collapse the <i>MotionManager</i> window.

The timeline area is divided by vertical grid lines corresponding to numerical markers showing the time. The numerical markers start at 00:00:00. You may click and drag a key to define the beginning or end time of the animation or motion simulation.

After a simulation is completed, you will see several horizontal bars appear in the timeline area. They are *Change bars* in *Motion*. *Change bars* connect key points. They indicate a change between key points, which characterize the duration of animation, view orientation, etc. More about the timeline area will be discussed in *Lesson 2*.

Switching back and forth between *Motion* and *SOLIDWORKS* is straightforward. All you have to do is click the *Model* tab (back to *SOLIDWORKS*) and *Motion Study* tab (to *Motion*) at the bottom of the graphics area.

### ***Defining Motion Entities***

The basic entities of a valid *Motion* simulation model consist of ground parts (or ground body), moving parts (or moving bodies), constraints (imposed by mates in *SOLIDWORKS* assembly), initial conditions (usually position and velocity of a moving body), and forces and/or drivers. Each of the basic entities will be briefly discussed next. More details can be found in later lessons.

#### *Ground Parts (or Ground Body)*

A ground part, or a ground body, represents a fixed reference in space. The first component brought into the assembly is usually stationary, therefore becoming a ground part. Parts (or sub-assemblies) assembled to the stationary components without any possibility to move become part of the ground body. A symbol (*f*) is placed in front of stationary components in the browser. You will have to identify moving and non-moving parts in your assembly, and use assembly mates to properly constrain the movement of the moving parts and completely fix the motion of non-moving parts.

#### *Moving Parts (or Bodies)*

A moving part or body is an entity representing a single rigid component (or link) that moves relatively to other parts (or bodies). A moving part may consist of a single *SOLIDWORKS* part or a sub-assembly composed of multiple parts. When a sub-assembly is designated as a moving part, none of its composing parts is allowed to move relative to one another within the sub-assembly.

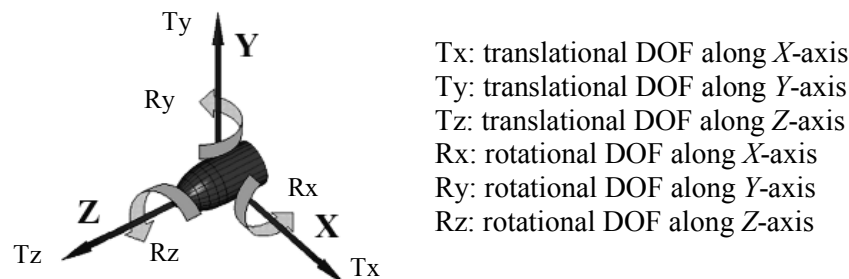


Figure 1-11 Translational and Rotational Degrees of Freedom

A moving body has six degrees of freedom: three translational and three rotational (shown in Figure 1-11), while a ground body has none. That is, a moving body can translate and rotate along the X-, Y-, and Z-axes of a coordinate system. Rotation of a rigid body is measured by referring the orientation of its

local coordinate system to the global coordinate system, which is shown at the lower left corner on the graphics area in *SOLIDWORKS*. In *Motion*, the local coordinate system is assigned automatically, usually at the mass center of the part or sub-assembly. Mass properties, including total mass, inertia, etc., are calculated using part geometry and material properties referring to the local coordinate system. A symbol (-) is placed in front of a moving component in the browser.

### Constraints

As mentioned earlier, an unconstrained rigid body in space has six degrees of freedom: three translational and three rotational. When you add a joint (or a constraint) between two rigid bodies, you remove degrees of freedom between them. Since *SOLIDWORKS* 2008, more commonly employed joints, such as revolute, translation, cylindrical joint, etc., in *Motion* have been replaced by assembly mates, such as coincident and concentric. Like joints, assembly mates remove degrees of freedom between parts.

Each independent movement permitted by a constraint (either a joint or an assembly mate) is a free degree of freedom. The free degrees of freedom that a constraint allows can be translational or rotational along the three perpendicular axes, either global or local. For example, a concentric mate between the propeller assembly and the case of a single piston engine shown in Figure 1-12 allows one translational DOF (movement along the center axis, in this case, the *X*-axis of the global coordinate system, called reference triad in *SOLIDWORKS*) and one rotational DOF (rotating along *X*-axis). Since the case assembly is stationary, serving as the ground body, the propeller assembly has two free DOFs. Adding a coincident mate between the two respective mating faces of the engine case and the propeller shown in Figure 1-12 removes the remaining translational DOF, yielding a desired assembly that resembles the physical situation; i.e., with only the rotational DOF (along the *X*-axis).

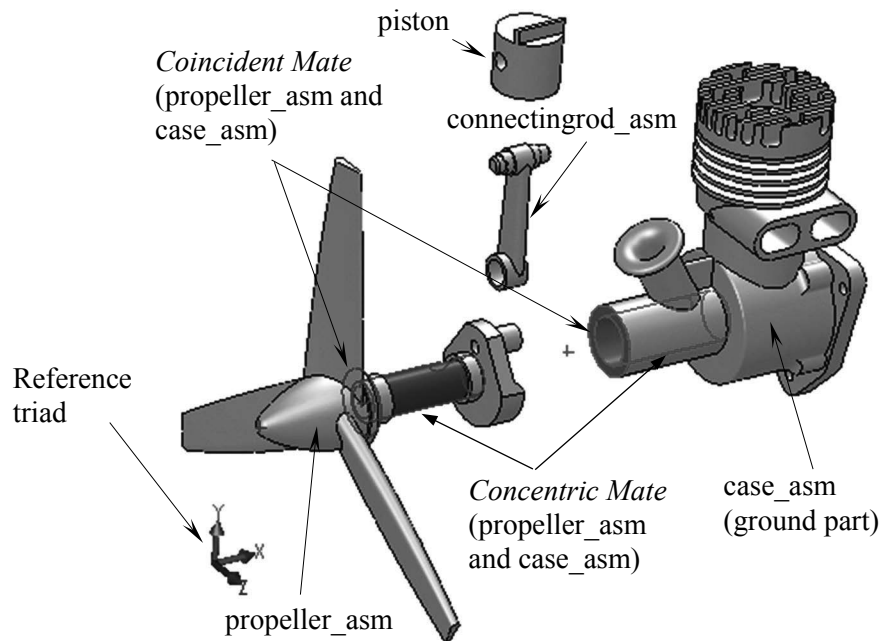


Figure 1-12 Assembly Mates Defined for the Propeller (Explode View)

In creating a motion model, instead of completely fixing all body motion, certain DOFs (translational and/or rotational) are left to allow for designated movement called free DOF. Such a movement will be either driven by a motor (for kinematic analysis) or determined by a force in dynamic analysis. For example, a rotary motor is created to drive the rotational DOF of the propeller in the engine

example. This rotary motor rotates the propeller at a prescribed angular velocity. In addition to velocity, you may use the motor to drive a DOF at a prescribed displacement or acceleration, either translation (using a linear motor) or rotational (using a rotary motor). You may apply a force or a torque to a free translational or rotational DOF, respectively, for a dynamic simulation.

It is extremely important to understand assembly mates in order to create working motion models that correctly simulate the physical behavior of the mechanism. In addition to standard mates such as concentric and coincident (see Table 1-2 for a list of *SOLIDWORKS* standard mates), *SOLIDWORKS* provides advanced mates and mechanical mates (see Tables 1-3 and 1-4, respectively). Advanced mates offer additional ways to constrain or couple movements between bodies. For example, a linear coupler allows the motion of a translational or a rotational DOF of a given mate to be coupled to the motion of another mate. Also, the rotation motion of the propeller may be coupled to the translational motion of the piston (certainly if it makes sense). A coupler removes one additional degree of freedom from the motion model. Also, path mate (one of the advanced mates in *SOLIDWORKS*) allows a part to move along a curve slot, a groove, or fluting, varying its moving direction specified by the path curve. Such a capability supports animation and motion analysis of a different set of applications involving curvilinear motion. We will learn path mate in *Lesson 9*.

Mechanical mates include cam follower, gear, hinge, rack and pinion, screw, and universal joint. These are essential for motion model yet extremely easy to create in *Motion*. Some of these mates will be further discussed in later lessons (for examples Lessons 10 to 12).

Table 1-2 Standard Mates of *SOLIDWORKS* Assembly









Standard Mates	Descriptions
Coincident 	Positions selected faces, edges, and planes (in combination with each other or combined with a single vertex) so they share the same infinite plane. Positions two vertices so they touch.
Parallel 	Places the selected items so they remain a constant distance apart from each other.
Perpendicular 	Places the selected items at a 90° angle to each other.
Tangent 	Places the selected items tangent to each other (at least one selection must be a cylindrical, conical, or spherical face).
Concentric 	Places the selections so that they share the same center line.
Lock 	Maintains the position and orientation between two components.
Distance 	Places the selected items with the specified distance between them.
Angle 	Places the selected items at the specified angle to each other.
Default	Place the first part to the default coordinate system in assembly.

Table 1-3 Advanced Mates of *SOLIDWORKS* Assembly














Advanced Mates	Descriptions
Symmetric 	Forces two similar entities to be symmetric about a plane or planar face.
Width 	Centers a tab within the width of a groove.
Path 	Constrains a selected point on a component to a path.
Linear/Linear Coupler 	Establishes a relationship between the translation of one component and the translation of another component.
Limit 	Allows components to move within a range of values for distance and angle mates.
Profile Center 	Automatically center-aligns geometric profiles to each other and fully defines the components.

Table 1-4 Mechanical Mates of *SOLIDWORKS* Assembly

Mechanical Mates	Descriptions
Cam 	Forces a cylinder, plane, or point to be coincident or tangent to a series of tangent extruded faces.
Gear 	Forces two components to rotate relative to one another about selected axes (or respective pitch circles).
Hinge 	Limits the movement between two components to one rotational degree of freedom.
Rack and Pinion 	Linear translation of one part (the rack) causes circular rotation in another part (the pinion), and vice versa.
Screw 	Constrains two components to be concentric, and also adds a pitch relationship between the rotation of one component and the translation of the other.
Slot 	Mates bolts to straight or arced slots and mate slots to slots. You can select an axis, cylindrical face, or a slot to create slot mates
Universal Joint 	The rotation of one component (the output shaft) about its axis is driven by the rotation of another component (the input shaft) about its axis.

In addition to mates, *Motion* provides solid body contact that helps to simulate physical problems involving contacts between bodies. Essentially, solid body contact applies a force to separate the parts when they are in contact and prevent them from penetrating each other. The solid body contact will become active as soon as the parts are touching each other. We will learn to create a contact constraint in *Lesson 3*.

### Degree of Freedom Count

As mentioned earlier, an unconstrained body in space has six degrees of freedom. When assembly mates are added to assemble parts, constraints are imposed to eliminate DOFs accordingly so as to restrict the relative motion between them as desired.

Let us go back to the engine example shown in Figure 1-12. A concentric mate between the propeller and the engine case restricts movement on four DOFs ( $T_y$ ,  $T_z$ ,  $R_y$ , and  $R_z$ ) so that only two movements are allowed, one translational ( $T_x$ ) and one rotational ( $R_x$ ). In order to restrict the translational movement, a coincident mate was added. A coincident mate between two respective mating faces of the propeller and the engine case restricts movement on three DOFs; i.e.,  $T_x$ ,  $R_y$ , and  $R_z$ . Even though combining these two mates gives desired rotational motion between the propeller and the case, there are redundant DOFs being restricted; i.e.,  $R_y$  and  $R_z$  in this case.

It is important that you understand how to count the overall degrees of freedom for a motion model. For a given motion model, the number of degrees of freedom can be determined by using the Gruebler's count, defined as:

$$D = 6M - N - O \quad (1.3)$$

where  $D$  is the Gruebler's count representing number of free degrees of freedom of the mechanism,  $M$  is the number of bodies excluding the ground body,  $N$  is the number of DOFs restricted by all mates, and  $O$  is the number of motion drivers (motors) defined in the system. For the motion model consisting of the propeller, the engine case, and the rotary motor, the Gruebler's count is:

$$D = 6 \times 1 - (4 + 3) - 1 = -2$$

However, we know that the propeller can only rotate along the  $X$ -axis; therefore, there is only one DOF for the system (Rx). The count should be 1. After adding the rotary motor, the count becomes zero. The calculation gives us  $-2$  due to the fact that there are two redundant DOFs,  $R_y$  and  $R_z$ , which were restrained by both concentric and coincident mates. If we remove the redundant DOFs, the count becomes:

$$D = 6 \times 1 - (4 + 3 - 2) - 1 = 0$$

Another example, a door assembled to door frame using two hinge joints. Each hinge joint allows only one rotational movement along the axis of the hinge. The second hinge adds five redundant DOFs. The Gruebler's count becomes:

$$D = 6 \times 1 - 2 \times 5 = -4$$

Again, if we remove the redundant DOFs, the count becomes, as it should be:

$$D = 6 \times 1 - (2 \times 5 - 5) = 1$$

This is before any motor is added.

For kinematic analysis, the Gruebler's count must be equal to or less than 0 after adding motors. The *ADAMS/Solver* recognizes and deactivates redundant constraints during analysis. For a kinematic analysis, if you create a model and try to animate it with a Gruebler's count greater than 0, the animation will not run and an error message will appear. For the door example, the vertical movement constrained by the second hinge is identified as redundant and removed from the solution. As a result, if conducting a dynamic simulation, the entire vertical force is carried by the first hinge. No (reaction) force will be calculated at the second hinge.

To get the Gruebler's count to zero, it is often required to replace mates that remove a larger number of constraints with mates that remove a smaller number of constraints and still restrict the mechanism motion in the same way. This is usually difficult if not entirely impossible. *Motion* detects the redundancies and ignores redundant DOFs in all analyses, except for dynamic analysis. In dynamic analysis, the redundancies lead to an outcome with a possibility of incorrect reaction results, yet the motion is correct. For complete and accurate reaction forces, it is critical that you eliminate redundancies from your mechanism if possible. The challenge is to find the mates that will impose non-redundant constraints and still allow for the intended motion. Very often this is not possible in *Motion* since instead of using regular joints (like revolute joint) it employs more primitive assembly mates for creating motion models. A concentric and a coincident mate in *SOLIDWORKS* is kinematically equivalent to a revolute joint, as illustrated in Figure 1-12 between the propeller and the engine case. A revolute joint removes five DOFs (with no redundancy); however, combining a concentric and a coincident mate removes seven DOFs, among which two are redundant. Using assembly mates to create motion models usually creates redundant DOFs.

The best strategy using *Motion* is to create an assembly that closely resembles the physical mechanism by using assembly mates that capture the characteristics of the motion revealed in the physical model. That is, first you should focus on creating an assembly that correctly captures the kinematic behavior of the mechanism. If you are conducting a dynamic analysis and want to capture reaction forces at critical components, you will have to examine the assembly mates and identify redundant DOFs. Then, check reaction forces at all mates for the relevant component and only take the reaction forces that make sense; mostly non-zero forces since zero reaction force is usually reported at redundant DOFs by *Motion*.

### Forces

Forces are used to operate a mechanism. Physically, forces are produced by motors, springs, dampers, gravity, etc. A force entity in *Motion* can be a force or torque. *Motion* provides three types of forces: applied forces, flexible connectors, and gravity.

Applied forces are forces that cause the mechanism to move in certain ways. Applied forces are very general, which involves defining the force direction and the force magnitude by specifying a constant force value or expression function, such as a harmonic function. The applied forces in *Motion* include action-only force or moment (where force or moment is applied at a point on a single rigid body, and no reaction forces are calculated), action and reaction force and moment, and impact force.

Flexible connectors resist motion and are simpler and easier to use than applied forces because you only supply constant coefficients for the forces, for instance a spring constant. The flexible connectors include translational springs, torsional springs, translational dampers, torsional dampers, and bushings.

A magnitude and a direction must be included for a force definition. You may select a predefined function, such as a harmonic function, to define the magnitude of the force or moment. For spring and damper, *Motion* automatically makes the force magnitude proportional to the distance or velocity, respectively, between two points, based on the spring constant and damping coefficient entered. The direction of a force (or moment) can be defined either along an axis defined by an edge or along the line between two points, where a spring or a damper is defined.

### Initial Conditions

In motion simulations, initial conditions consist of initial configuration of the mechanism and initial velocity of one or more components of the mechanism. Motion simulation must start with a properly assembled solid model that determines an initial configuration of the mechanism, composed by position and orientation of individual components. The initial configuration can be completely defined by assembly mates. However, one or more assembly mates will have to be suppressed, if the assembly is fully constrained, to provide adequate movement. In *SOLIDWORKS Motion*, initial velocity is defined as part of the definition of a moving part. The initial velocity can be translational or rotational along a direction spanned by the three axes.

### Motion Drivers

Motion drivers (or motors) are used to impose a particular movement on a free DOF over time. A motion driver specifies position, velocity, or acceleration as a function of time, and can control either translational or rotational motion. When properly defined, motion drivers will account for the remaining DOFs of the mechanism that bring the Gruebler's count to zero (exactly zero after removing all redundant DOFs) or less than zero for a kinematic analysis.

### Motion Simulation

The *ADAMS/Solver* employed by *Motion* is capable of solving general problems, such as static (equilibrium configuration), kinematic, and dynamic. Three numerical solvers are provided. They are GSTIFF, SI2\_GSTIFF, and WSTIFF. GSTIFF is the default (numerical) integrator and is fast and accurate for displacements. It is used for a wide range of motion simulations. SI2\_GSTIFF provides better accuracy of velocities and accelerations but can be significantly slower. WSTIFF provides better accuracy for special problems, such as discontinuous forces.



Static analysis is used to find the rest position (equilibrium condition) of a mechanism, in which none of the bodies are moving. A simple example of the static analysis is illustrated in Figure 1-13, in which an equilibrium position of the block is to be determined according to its own mass  $m$ , the two spring constants  $k_1$  and  $k_2$ , and the gravity  $g$ .

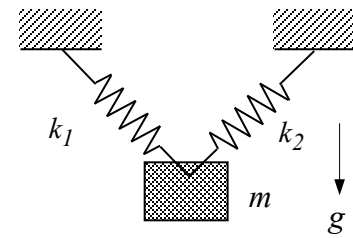


Figure 1-13 Static Analysis

As discussed earlier, kinematics is the study of motion without regard for the forces or torque. A mechanism can be driven by a motion driver for a kinematic analysis, where the position, velocity, and acceleration of each link of the mechanism can be analyzed at any given time. Figure 1-14 shows a servomotor driving a slider-crank mechanism at a constant angular velocity.

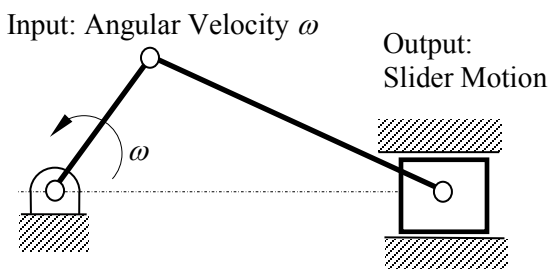


Figure 1-14 Kinematic Analysis

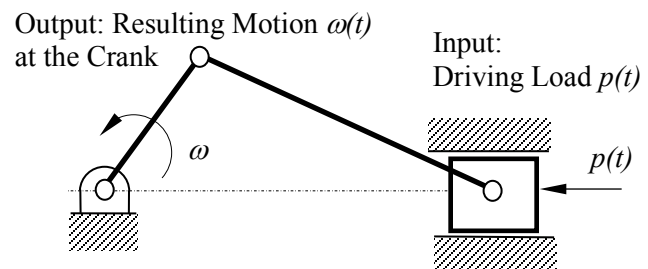


Figure 1-15 Dynamic Analysis

Dynamic analysis is employed for studying the mechanism motion in response to loads, as illustrated in Figure 1-15, in which a force  $p(t)$  is applied at the slider in the horizontal direction to cause the slider to move, leading to the movement of the connecting rod and the crank. This is the most complicated and common, and usually a more time-consuming analysis.

### Viewing Results

In *Motion*, results of the motion analysis can be visualized using animations, graphs, reports, and queries. Animations show the configuration of the mechanism in consecutive time frames. Animations will give you a global view on how the mechanism behaves; for example, the single-piston engine shown in Figure 1-16. You may also export the animation to AVI for various purposes.

In addition, you may choose a joint (assembly mates in *Motion*) or a part to generate result graphs; for example, position vs. time of the mass center of the piston in the engine example shown in Figure 1-17. These graphs give you a quantitative understanding on the characteristics of the mechanism.

You may also query the results by moving the mouse pointer closer to the curve in a graph and leave the pointer for a short period. The result data will appear next to the pointer. In addition, you may ask *SOLIDWORKS Motion* for a report that includes a complete set of results output in the form of textual data or a Microsoft® Excel spreadsheet.

In addition to the capabilities discussed above, *Motion* allows you to check interference between bodies during motion (please see *Lesson 7* for more details). Furthermore, the reaction forces calculated can be used to support structural analysis using, for example, *SOLIDWORKS Simulation*.

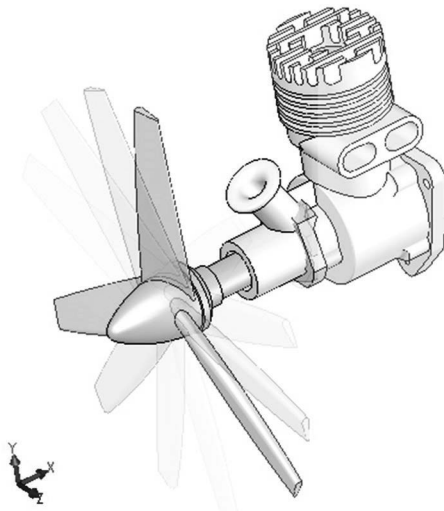


Figure 1-16 Motion Animation

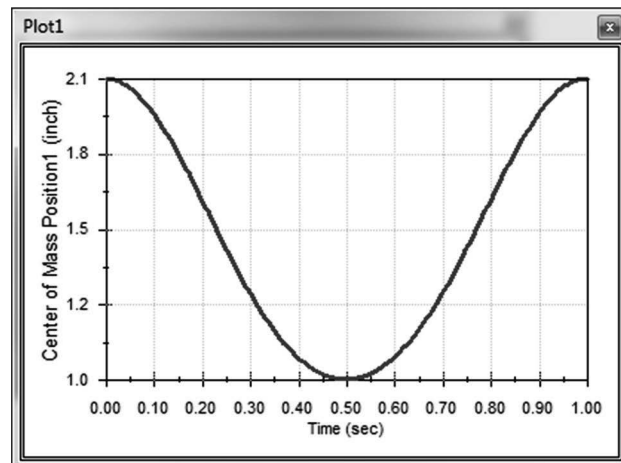


Figure 1-17 Graph of Position of the Piston vs. Time

## 1.5 Open Lesson 1 Model

A motion model for the single piston engine example shown in Figure 1-1 has been created for you. Download the files from [www.sdcpublishings.com](http://www.sdcpublishings.com), unzip them, and locate the engine model (filename: *Lesson1withResults.SLDASM*) under *Lesson 1*. Copy or move *Lesson 1* folder to your hard drive.

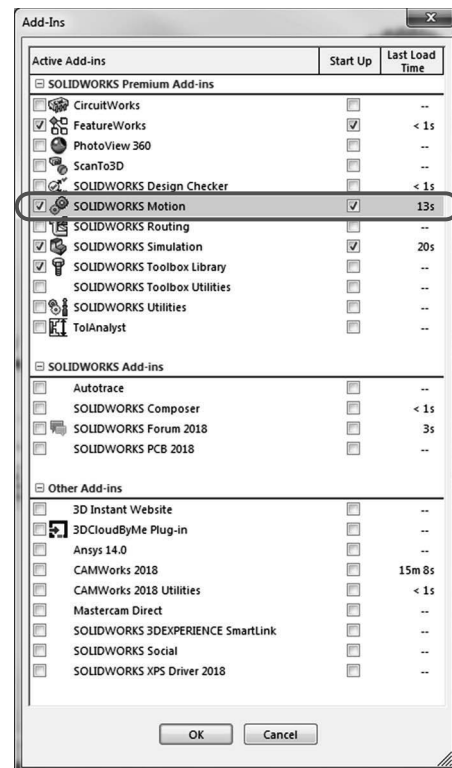
Start *SOLIDWORKS* and open assembly *Lesson1withResults.SLDASM*. You should see an assembled engine model similar to that of Figure 1-1. Also, the *Motion Study* tab (with a default name *Motion Study 1*) will appear at the bottom of the graphics area.

To use full capabilities of *Motion*, you will need to activate the *Motion* module by choosing from the pull-down menu

*Tools* → *Add-Ins*

In the *Add-Ins* window shown in Figure 1-18, click *SOLIDWORKS Motion* in both boxes (*Active Add-ins* and *Start Up*), and then click *OK*.

Click the *Motion Study* tab to bring up the *MotionManager* window. Note that a rotary motor, *RotaryMotor1*, has been added to the motion model. Click the *Play* button on the *Motion* toolbar; you should see that the propeller starts rotating, similar to that of Figure 1-16. Also, you may want to hide *case\_asm* to see the motion inside the case, especially between the connection rod and the piston. You may do so by right clicking *case\_asm<1>* from the *MotionManager* tree and choosing *Hide*.

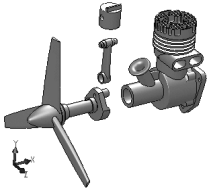
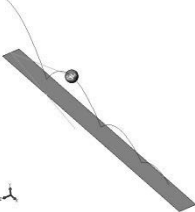
Figure 1-18 The *Add-Ins* Window

## 1.6 Motion Examples

More motion examples will be introduced in this book to illustrate the step-by-step details of modeling, simulation, and result visualization capabilities in *Motion*.

We will start with the same engine example in *Lesson 2*. *Lesson 2* will introduce numerous animations and basic motion study capabilities, with the focus that helps you become familiar with the basic operations in *MotionManager* window, especially the timeline area. In *Lesson 3*, a simple ball-throwing example will be presented. This example provides you a quick run-through on using *Motion*. This lesson will also involve creating a solid body contact. *Lessons 4* through *12* focus on modeling and analysis of basic mechanisms and dynamic systems. In these lessons, you will learn assembly mates; forces and connections, including springs, gears, cam-followers; drivers and forces; various analyses; and graphs and results. Two application lessons are offered. *Lesson 13* presents a motion study of riding object on a recreational waterslide. *Lesson 14* illustrates the kinematic analysis of suspension of a SAE Formula style racecar. All examples and topics to be discussed in each lesson are summarized in the following table.

Table 1-5 Examples Employed in This Book

Lesson	Title	Motion Model	Problem Type	Things to Learn
2	Single Piston Engine Example		Animation and basic motion	<ol style="list-style-type: none"> <li>1. This lesson focuses on helping you become familiar with the basic operations in <i>MotionManager</i> window.</li> <li>2. You will learn animation and basic motion study capabilities, especially using <i>Animation Wizard</i> to create simple animations in assembly.</li> </ol>
3	Ball Throwing Example		Particle Dynamics	<ol style="list-style-type: none"> <li>1. This lesson offers a quick run-through of general modeling and simulation capabilities in <i>Motion</i>.</li> <li>2. You will learn the general process of using <i>Motion</i> to construct a motion model, run simulation, and visualize motion simulation results.</li> <li>3. Gravity will be turned on and a solid body contact will be defined between the ball and the ground to simulate a ball bouncing scenario.</li> <li>4. Simulation results are verified using analytical equations of motion.</li> </ol>


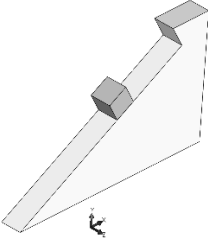
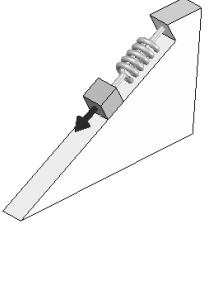

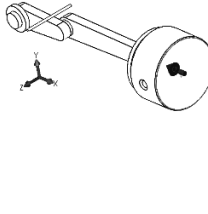
4	Simple Pendulum		Particle Dynamics	<ol style="list-style-type: none"> <li>1. This lesson provides a discussion on using concentric and coincident mates to create a revolute joint that allows the simulation of pendulum motion using <i>Motion</i>.</li> <li>2. Simulation results are verified using analytical equations of motion.</li> </ol>
---	-----------------	---	-------------------	---

Table 1-5 Examples Employed in This Book (Cont'd)

Lesson	Title	Motion Model	Problem Type	Things to Learn
5	Friction of a Sliding Object		Particle Dynamics	<ol style="list-style-type: none"> <li>1. This lesson provides a discussion on friction of a sliding object.</li> <li>2. Both contact friction and joint friction offered in <i>Motion</i> are presented.</li> <li>3. Simulation results are verified using analytical equations of motion.</li> </ol>
6	Spring-Mass System		Particle Dynamics	<ol style="list-style-type: none"> <li>1. This is a classical spring-mass system example you learned in Sophomore Dynamics class.</li> <li>2. You will learn how to create a mechanical spring, align the block with the slope surface, and add an external force to pull the block.</li> <li>3. Simulation results are verified using analytical equations learned in Dynamics.</li> </ol>
7	Slider Crank Mechanism		Multibody Dynamic Analyses	<ol style="list-style-type: none"> <li>1. This lesson uses a slider-crank mechanism to conduct kinematic and dynamic simulations.</li> <li>2. You will learn to create motion driver as well as add a firing force for simulation.</li> <li>3. The interference checking capability will be discussed.</li> <li>4. Kinematic analysis results are verified using analytical equations of motion.</li> </ol>
8	Event-Based Dynamic Simulation		Multibody Dynamic Analyses	<ol style="list-style-type: none"> <li>1. This lesson uses the same slider-crank mechanism to present event-based dynamic simulation.</li> <li>2. You will learn to create task trigger and task action, from which a more realistic firing load is created for the simulation of the slider-crank mechanism.</li> </ol>


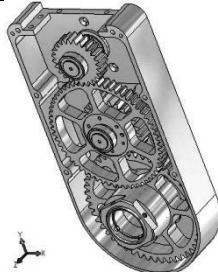
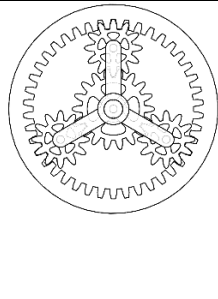
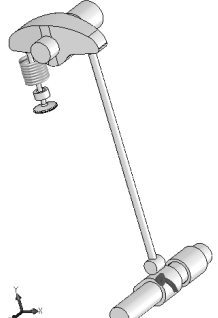
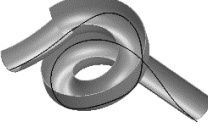

9	Rail-Carriage Example		Path Mate	<ol style="list-style-type: none"> <li>1. This lesson introduces a useful and powerful mate provided in <i>SOLIDWORKS</i>, the path mate. Such a capability supports animation and motion study of a different set of applications involving curvilinear motion.</li> </ol>
---	-----------------------	---	-----------	---

Table 1-5 Examples Employed in This Book (Cont'd)

Lesson	Title	Motion Model	Problem Type	Things to Learn
10	Compound Spur Gear Train		Gear Train Analysis	<ol style="list-style-type: none"> <li>1. This lesson focuses on simulating motion of a spur gear train system.</li> <li>2. You will learn how to use <i>SOLIDWORKS</i> create gear mates, and use <i>Motion</i> to simulate the gear train motion.</li> <li>3. Gear rotation speed is verified using analytical calculation.</li> </ol>
11	Planetary Gear Train Systems		Gear Train Analysis	<ol style="list-style-type: none"> <li>1. This lesson focuses on simulating motion of planetary gear train systems.</li> <li>2. You will learn how to use <i>SOLIDWORKS</i> create gear mates, and use <i>Motion</i> to simulate the gear train motion.</li> <li>3. Both single planetary and multi-planetary gear train systems are included.</li> <li>4. Gear rotation speed is verified using analytical calculations.</li> </ol>
12	Cam and Follower		Multibody Dynamic Analysis	<ol style="list-style-type: none"> <li>1. This lesson discusses cam and follower connection.</li> <li>2. An inlet or outlet valve system of an internal combustion engine will be created and simulated.</li> <li>3. Position and velocity of the valve will be graphed to monitor the motion of the system as well as assess the engineering design of the system.</li> </ol>

13	Simulation of Recreational Waterslide		Single-body Dynamic Analysis	<ol style="list-style-type: none"><li>1. This lesson presents a real-life application, simulation of riding object on a recreational waterslide using <i>Motion</i>.</li><li>2. A brief discussion on an analytical calculation method and experimental validation of the method is offered.</li><li>3. Discussion on a few modeling considerations in using <i>Motion</i> is also provided.</li></ol>
14	Quarter suspension of a racecar		Multibody Kinematic Analysis	<ol style="list-style-type: none"><li>1. This lesson discusses kinematic analysis of a quarter racecar suspension, including the addition of a cam representing the road profile.</li><li>2. Important kinematic measures of suspension design, including shock travel and camber angle, are included in discussion.</li></ol>