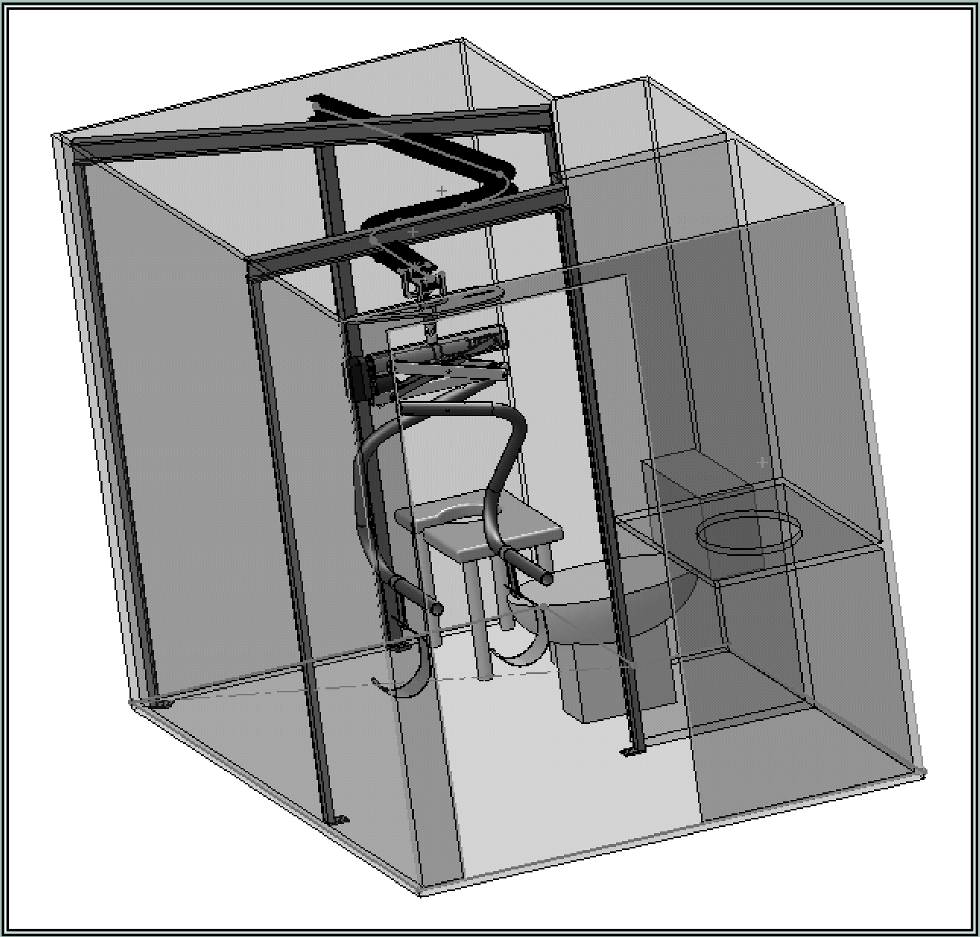
Motion Simulation and Mechanism Design

with SolidWorks Motion 2009



Kuang-Hua Chang, Ph.D.

School of Aerospace and Mechanical Engineering

The University of Oklahoma

SDC

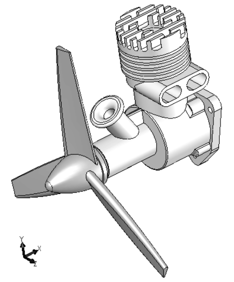
PUBLICATIONS

Schroff Development Corporation

www.schroff.com

Better Textbooks. Lower Prices.

Lesson 1: Introduction to *SolidWorks Motion*



1.1 Overview of the Lesson

This lesson intends to provide you with a brief overview of *SolidWorks Motion*. *SolidWorks Motion*, formerly called *COSMOSMotion* (*SolidWorks* 2008 and before), is a virtual prototyping tool that supports animation and analysis of motion, 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. 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 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.

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 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 *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 follows:

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 design 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 of 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 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.

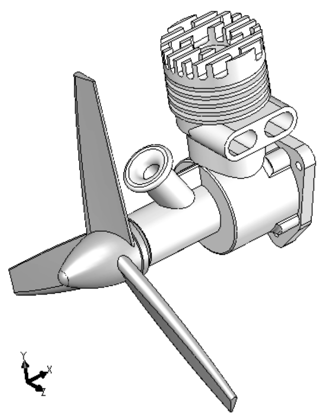


Figure 1-1 A Single Piston Engine (Unexploded View)

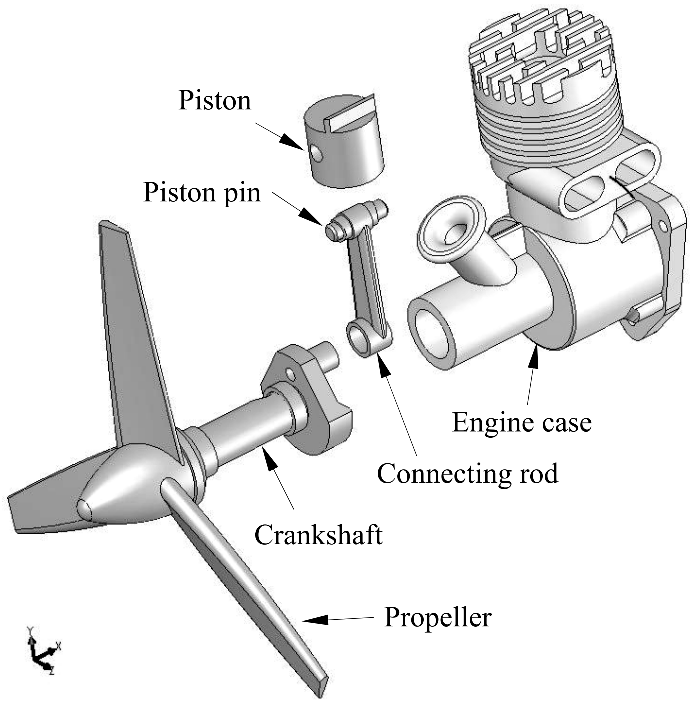
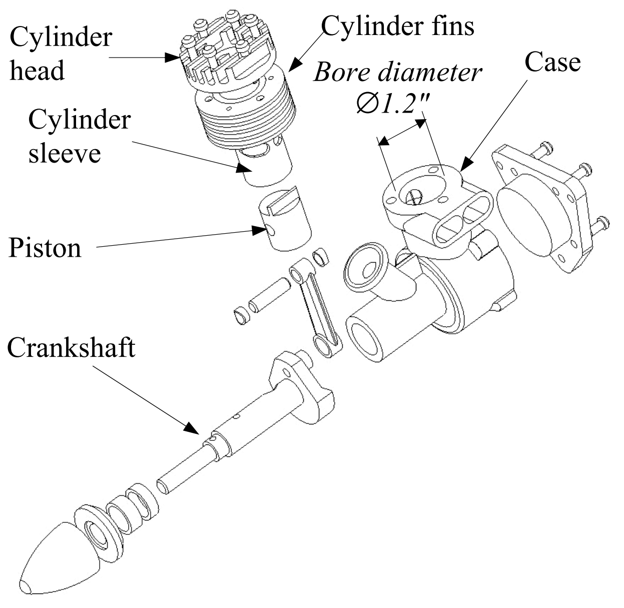


Figure 1-2 Single Piston Engine (Explode View)

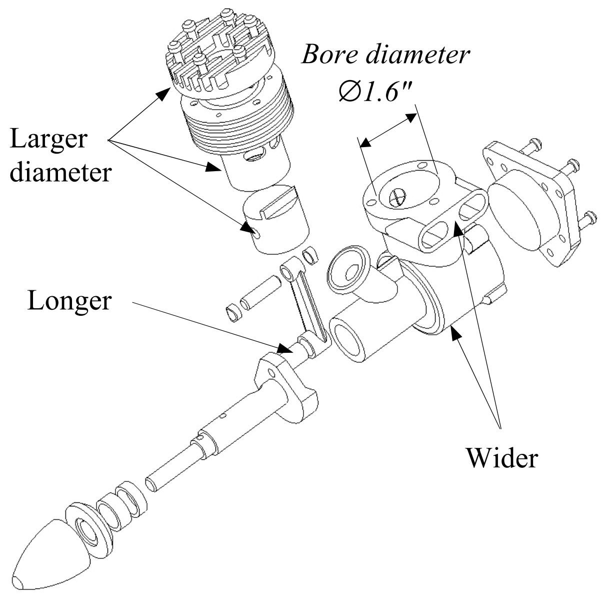
The capabilities available in *Motion* also help you search for better design alternatives. A better design alternative is very much problem dependent. It is critical that a design problem be clearly defined by the designer up front before searching for better design alternatives. For the engine example, a better alternative can 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 mechanical system. 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 to all parts affected automatically. 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 instance, in this engine example, a change in the bore diameter of the engine case will alter not only the geometry of the engine case itself, 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.



(a) Bore Diameter 1.2"



(b) Bore Diameter 1.6"

Figure 1-3 A Single-Piston Engine – Exploded 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. 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 analysis purpose. For example, a slider-crank mechanism represents the engine motion, as shown in Figure 1-5, which is a closed loop mechanism.

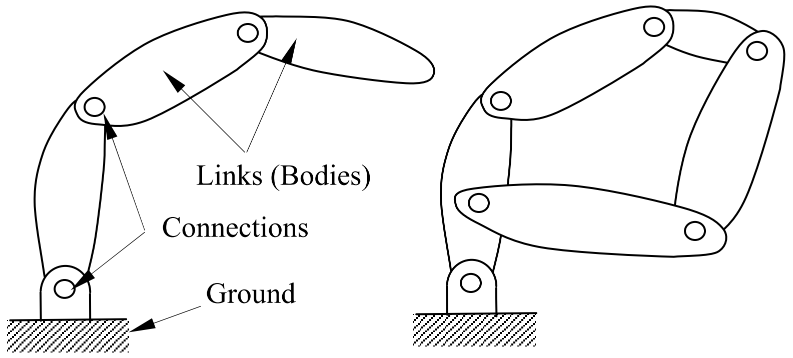
In general, there are two types of motion problems that you will have to solve in order to answer questions regarding mechanism analysis and design: kinematic and dynamic.

Kinematics is the study of motion without regard for the forces that cause the motion. A kinematic mechanism must 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 must be conducted before 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 the 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,

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

where $(\dot{\ })$ 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

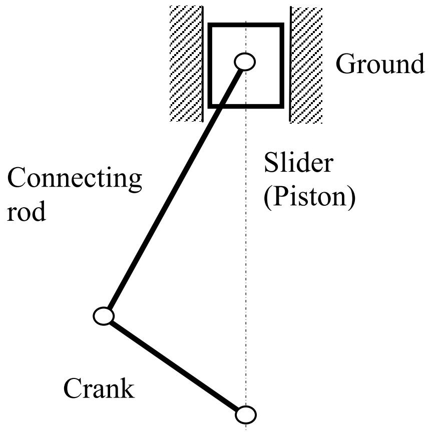


Figure 1-5 Schematic View of the Engine Motion Model

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,

$$\sum M = -mg\ell \sin\theta = I\ddot{\theta} = m\ell^2\ddot{\theta} \qquad (1.2)$$

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

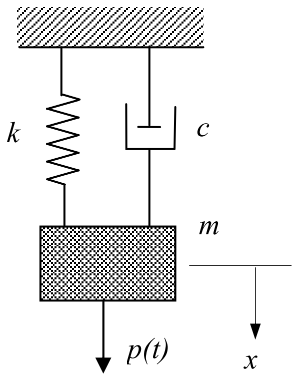


Figure 1-6 The Spring-Mass-Damper System

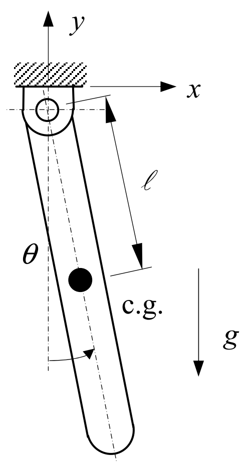


Figure 1-7 A Simple Pendulum

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 every single body in the system at all time. 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 loads 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, position, velocity, and acceleration results are identical to those of kinematic analysis. However, 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 servo motors 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.

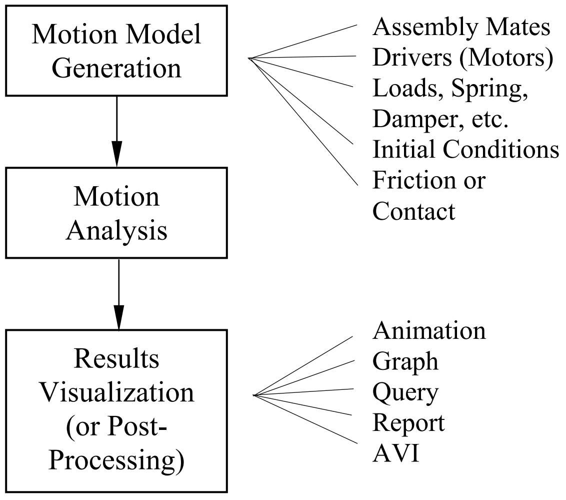


Figure 1-8 General Process of Using *SolidWorks Motion*

The analysis or simulation capabilities in *Motion* employ simulation engine, *ADAMS/Solver*, which solves the equations of motion for your 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 file portability.

Operation Mode

Motion is embedded in *SolidWorks*. It is indeed an add-on module of *SolidWorks*, and transition from *SolidWorks* to *Motion* is seamless. All the solid models, 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 checking as well as detecting contact between bodies for the mechanism during simulation in *Motion*.

User Interfaces – MotionManager

User interface of the *Motion* is embedded in and identical to that of *SolidWorks*. *SolidWorks* users should find it is straightforward to maneuver in *Motion*. The main interface for using *Motion* is through *MotionManager*, 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.

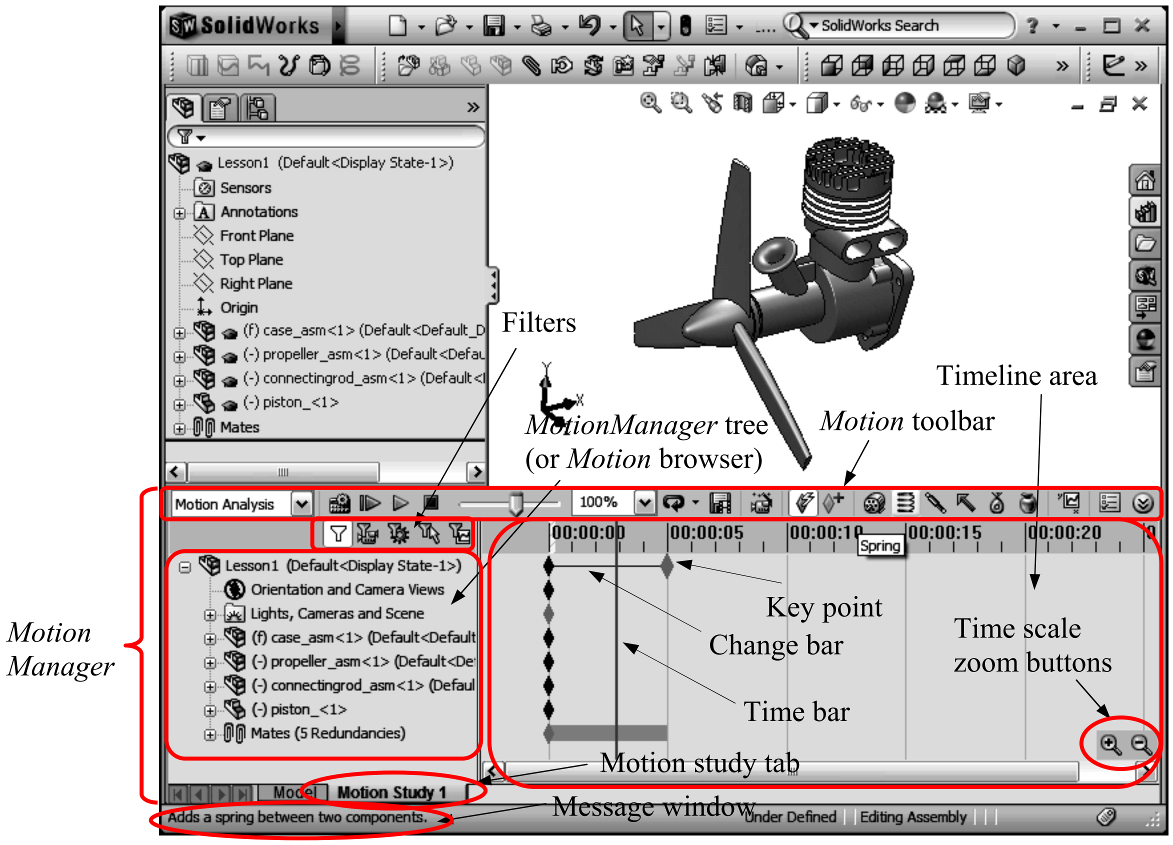


Figure 1-9 User Interface of *SolidWorks Motion* – *MotionManager*

As shown in Figure 1-9, the user interface window of *MotionManager* consists of the *MotionManager* tree (or *Motion* browser), the *Motion* toolbar, filters, timeline area, etc.

MotionManager Tree (Motion Browser)

Components are mapped from the *SolidWorks* assembly into the *MotionManager* tree automatically, including root assembly, parts, sub-assemblies, and mates. A single *Orientation and Camera Views* entity is also added. This entity will help you create precise animation as desired, but has less to do with the motion simulation. Each part and sub-assembly entity 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 resulting branch will be added to the tree once the motion analysis is completed and result graphs are created. Similar to *SolidWorks*, right clicking on a node in the *MotionManager* tree will bring up command options that you can choose to modify or adjust the entity.

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 options, saving options, *Animation Wizard*, key point controls, and simulation elements. As you move the mouse pointer over a button, a brief description about the functionality of the button will appear. 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 lower left corner, as shown in Figure 1-9. Click some of the buttons and try to become more familiar with their functions. The buttons in *Motion* tool bar and their functions are also summarized in Table 1-1 with more details.

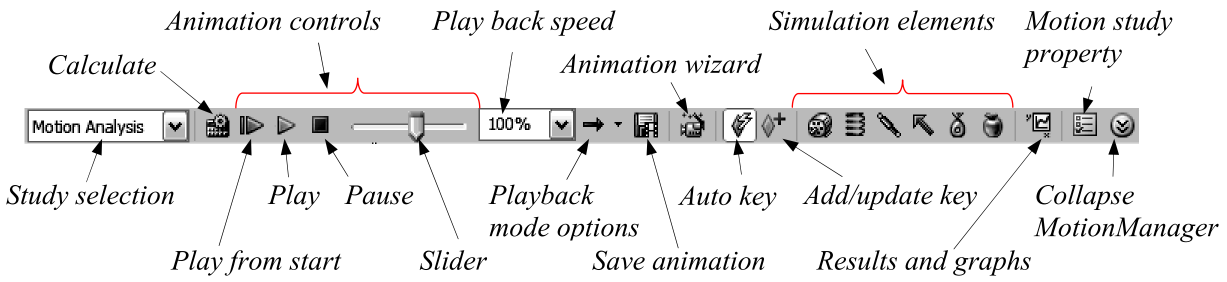


Figure 1-10 The *Motion* Toolbar

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 provides you accurate simulation and analysis on the effects of motion elements (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* option to graph 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 account. *Motion Analysis* is the most computationally intensive of the three options. The better your understanding of the physics of the motion you require, the better your results. You can use *Motion Analysis* to run impact analysis studies to understand component response to different types of forces.

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 and ready for your use.

Table 1-1 The Shortcut 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	Continuous play, from beginning to end then loop to beginning and continue playing.
↔	Playback mode: Reciprocal	Continuous 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	Calculate results and graphs.
	Motion study properties	Define motion study solution parameters.
	Collapse *MotionManager*	Collapse the *MotionManager* window.

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. 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. *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 to 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.

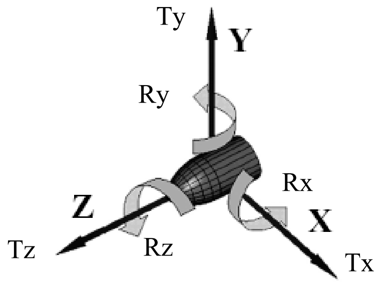
Ground Parts (or Ground Body)

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 the stationary components in the browser. You will have to identify moving and non-moving parts in your assembly, and use assembly mates to completely constrain the movement of the non-moving parts.

Moving Parts (or Bodies)

A moving part or body is an entity represents 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.

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.



Tx: translational dof along *X*-axis
Ty: translational dof along *Y*-axis
Tz: translational dof along *Z*-axis
Rx: rotational dof along *X*-axis
Ry: rotational dof along *Y*-axis
Rz: rotational dof along *Z*-axis

Figure 1-11 Translational and Rotational Degrees of Freedom

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. More discussion on joints can be found in Appendix A.

However, in *Motion 2009*, more commonly employed joints, such as revolute, translation, cylindrical joint, etc., have been replaced by assembly mates only. Like joints, assembly mates remove degrees of freedom between parts.

Each independent movement permitted by a constraint (either a joint or 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. 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) 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 dof's. By adding a coincident mate between the two respective 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.

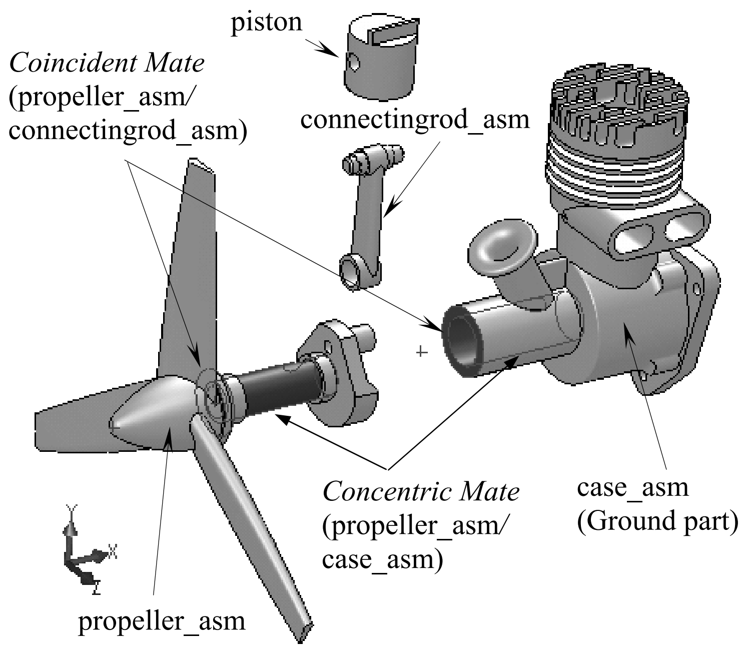


Figure 1-12 Mates Defined for the Engine Model (Exploded View)

In creating a motion model, instead of completely fixing all the movements, certain dof's (translational and/or rotational) are left to be either driven by a motor (for kinematic analysis) or determined by a force via dynamic analysis. For example, an angular motor is created to drive the rotational dof of the propeller in the engine example. This motor rotates the propeller at a prescribed angular velocity. In addition to prescribed 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 rotational motor).

It is extremely important for you to understand assembly mates in order to create successful motion models. In addition to standard mates such as concentric and coincident, *SolidWorks* provides advanced mates and mechanical mates. Advanced mates provide 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. 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 whole new set of applications involves curvilinear motion. We will learn path mate in *Lesson 7*.

The 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.

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

Degrees of Freedom

As mentioned earlier, an unconstrained body in space has six degrees of freedom; i.e., three translational and three rotational. When mates are added to assemble parts, constraints are imposed to restrict the relative motion between them.

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 dof's (Ty, Tz, Ry, and Rz) so that only two movements are allowed, one translational (Tx) and one rotational (Rx). In order to restrict the translational movement, a coincident mate was added. A coincident mate between two respective faces of the propeller and the engine case restrict movement on three dof's; i.e., Tx, Ry, and Rz. Even though combining these two mates gives desired rotational motion between the propeller and the case, there are redundant dof's being restricted; i.e., Ry and Rz 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:

$$D = 6M - N - O \tag{1.3}$$

where D is the Gruebler's count representing the free degrees of freedom of the mechanism, M is the number of bodies excluding the ground body, N is the number of dof's 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 (Rx) for the system. 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 dof's, Ry and Rz, which were restrained by both concentric and coincident mates. If we remove the redundant dofs, the count then 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 dof's. 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 large number of constraints with mates that remove a smaller number of constraints and still restrict the mechanism motion in the same way. *Motion* detects the redundancies and ignores redundant dof's 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. 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 2009* since instead of using regular joints (like revolute joint) it employs more primitive assembly mates for motion models. A concentric and a coincident mate 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 dof's (with no redundancy), however, combining a concentric and a coincident mate remove seven dof's, among which two are redundant. Using assembly mates to create motion models almost guarantees redundant dof's.

The best strategy using *Motion 2009* is to create an assembly closely resembles the physical mechanism by using mates that captures 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 exam the assembly mates carefully and identify redundant dof's. Then, check reaction forces at all mates for the interested component and only take the reaction forces that make sense to you; mostly non-zero forces (zero reaction force is reported at the redundant dof's in *Motion*). Also, you may show analysis message by selecting *Show all Motion Analysis messages* in the *Motion Study Properties* dialog box. In the analysis message, redundant dof's removed during analysis are listed. More details can be found in Appendix A.

Forces

Forces are used to operate a mechanism. Physically, forces are produced by motors, springs, dampers, gravity, etc. In addition, 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, but you must define 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 between two points, based on the spring constant and damping coefficient entered, respectively. The direction of a force (or moment) can be defined by 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 *Motion*, initial velocity is defined as part of definition for a moving part. The initial velocity can be translational or rotational.

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 dof's of the mechanism that brings the Gruebler's count to zero (exactly zero after removing all redundant dof's) or less then zero (with redundant dof) for a kinematic analysis.

Motion Simulation

The *ADAMS/Solver* employed by *Motion* is capable of solving typical engineering problems, such as static (equilibrium configuration), kinematic, and dynamic, etc. Three numerical solvers are provided. They are *GSTIFF*, *SI2_GSTIFF*, and *WSTIFF*. *GSTIFF* is the default integrator, and is fast and accurate for displacements. It is used for 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. More discussion about the solver algorithms will be discussed in *Lesson 7*.

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*.

As discussed earlier, kinematics is the study of motion without regard for the forces that cause the motion. 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 drives a mechanism at a constant angular velocity.

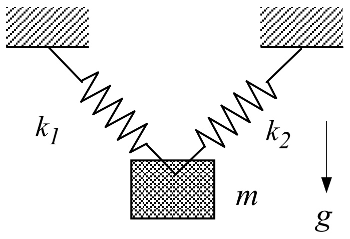


Figure 1-13 Static Analysis

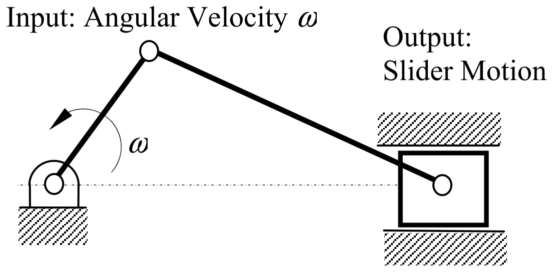


Figure 1-14 Kinematic Analysis

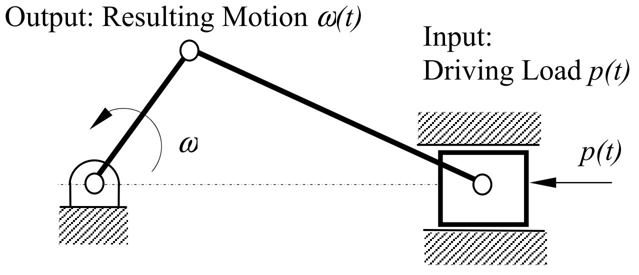

Output: Resulting Motion $\omega(t)$
Input: Driving Load $p(t)$
ω
$p(t)$

Figure 1-15 Dynamic Analysis

Dynamic analysis is employed for studying the mechanism motion in response to loads, as illustrated in Figure 1-15. This is the most complicated and commonly encountered in practices, and usually a more time-consuming analysis.

Viewing Results

In *Motion*, results of the motion analysis can be realized 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.

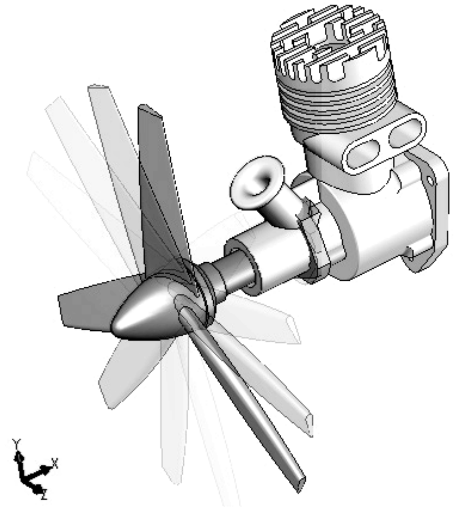


Figure 1-16 Motion Animation

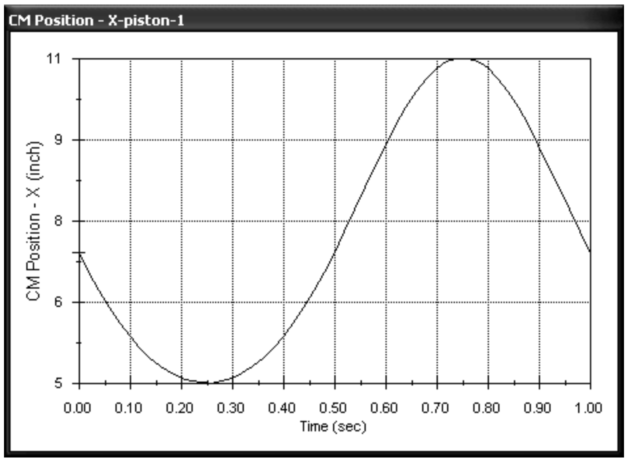


Figure 1-17 Graph of Position of the Piston vs. Time

In addition, you may choose a joint (in *Motion 2009*, assembly mate) or a part to generate result graphs, for example, the position vs. time 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 motion.

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 *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 6* for more details). Furthermore, the reaction forces calculated can be used to support structural analysis using, for example, *COSMOSWorks*.

1.5 Open Lesson 1 Model

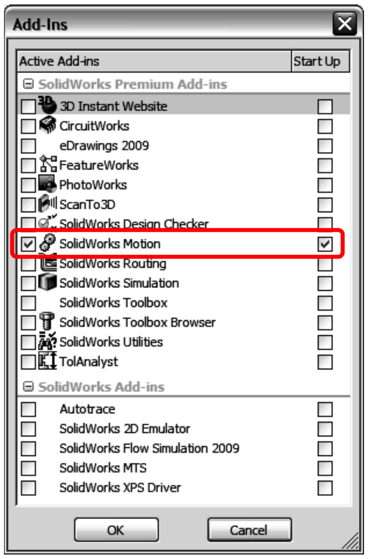
A motion model for the single piston engine model shown in Figure 1-1 has been created for you. Download the files from www.schroff1.com, unzip them, and locate the engine model under *Lesson 1*. Copy *Lesson 1* folder to your hard drive.

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. If you do not see this tab, you may not have activated the *Motion* add-in module. To activate the *Motion* module, choose from the pull-down menu

Tools > Add-Ins

In the *Add-Ins* window shown in Figure 1-18 click *SolidWorks Motion* in both boxes, 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 in the *Motion* toolbar; you should see that the propeller starts rotating, similar to that of Figure 1-16. **A**lso, you may want to hide *case_asm* to see the motion inside the case, especially between the connecting rod and the piston.


Add-Ins

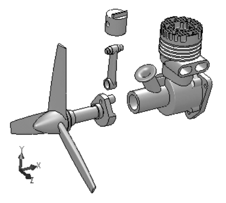
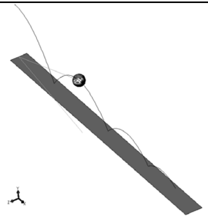
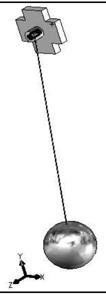
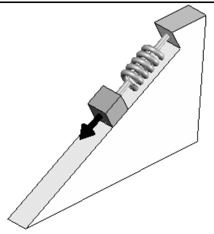
Active Add-ins	Start Up
SolidWorks Premium Add-ins	
☐ 3D Instant Website	☐
☐ CircuitWorks	☐
☐ eDrawings 2009	☐
☐ FeatureWorks	☐
☐ PhotoWorks	☐
☐ ScanTo3D	☐
☐ SolidWorks Design Checker	☐
☑ SolidWorks Motion	☑
☐ SolidWorks Routing	☐
☐ SolidWorks Simulation	☐
☐ SolidWorks Toolbox	☐
☐ SolidWorks Toolbox Browser	☐
☐ SolidWorks Utilities	☐
☐ TolAnalyst	☐
SolidWorks Add-ins	
☐ Autotrace	☐
☐ SolidWorks 2D Emulator	☐
☐ SolidWorks Flow Simulation 2009	☐
☐ SolidWorks MTS	☐
☐ SolidWorks XPS Driver	☐

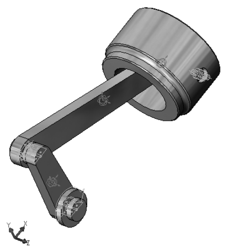
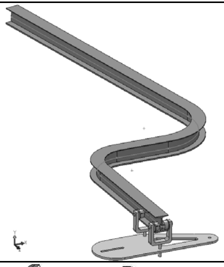
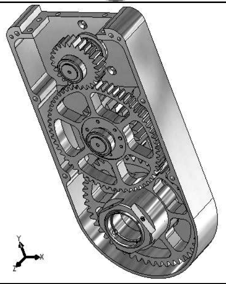
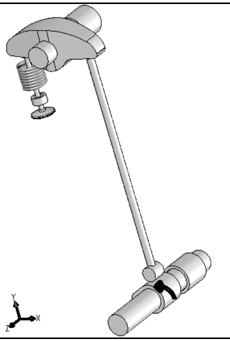
Figure 1-18 The *Add-Ins* Window

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*. This lesson will introduce numerous animations and basic motion study capabilities, with the focus on helping you become familiar with the basic operation in *MotionManager*, especially the timeline area. In *Lesson 3*, a simple ball-throwing example will be presented. This example will give you a quick run-through on using *Motion*. This lesson will also involve a 3D contact constraint. *Lessons 4* through *9* 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. All examples and main topics to be discussed in each lesson are summarized in the following table.

Table 1-2 Examples Employed in This Book

Lesson	Title	Motion Model	Problem Type	Things to Learn
2	Animations and Basic Motion – Single Piston Engine Example		Animation and basic motion	1. This lesson focuses on helping you become familiar with the basic operation in *MotionManager*. 2. You will learn animation and basic motion study capabilities, especially using *Animation Wizard* to create simple animations in assembly.
3	Ball Throwing Example		Particle Dynamics	1. This lesson offers a quick run-through of general modeling and simulation capabilities in *Motion*. 2. You will learn the general process of using *Motion* to construct a motion model, run simulation, and visualize the motion simulation results. 3. Gravity will be turned on and a 3D contact joint will be defined between the ball and the ground to simulate the ball bouncing scenario. 4. Simulation results are verified using analytical equations of motion.
4	Simple Pendulum		Particle Dynamics	1. This lesson provides more in depth about using concentric and coincident mates to create a pendulum in *Motion*. 2. Simulation results are verified using analytical equations of motion.
5	Spring-Mass System		Particle Dynamics	1. This is a classical spring-mass system example you learned in sophomore *Dynamics* class. 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. 3. Friction will be added between the block and the slope face. 4. Simulation results are verified using analytical equations learned in *Dynamics*.

6	Slider Crank Mechanism		Multibody Dynamic Analyses	1. This lesson uses a slider-crank mechanism to conduct kinematic and dynamic simulations. 2. You will also learn to create motion driver as well as add a firing force for simulation. 3. The interference checking capability will be discussed. 4. Kinematic analysis results are verified using analytical equations of motion.
7	Rail-Carriage Example		Path mate	1. This lesson introduces a very useful and powerful mate, the path mate. Such a capability supports animation and motion analysis of a whole new set of applications involves curvilinear motion.
8	Compound Spur Gear Train		Gear Train Analysis	1. This lesson focuses on simulating motion of a spur gear train system. 2. You will learn how to use *SolidWorks* and *Motion* to create a gear connection, and simulate the gear train motion. 3. Gear rotation speed are verified using analytical equations.
9	Cam and Follower		Multibody Dynamic Analysis	1. This lesson discusses cam and follower connection. 2. An inlet or outlet valve system of an internal combustion engine will be created and simulated. 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.