Mechanism Design and Analysis
Using PTC® Creo® Mechanism 3.0

Kuang-Hua Chang, Ph.D.
Visit the following websites to learn more about this book:
1.1 Overview of the Lesson

The purpose of this lesson is to provide you with a brief overview on PTC Creo Mechanism software tool or simply Mechanism. Mechanism is a virtual prototyping tool that supports mechanism design and analysis, which is also called motion analysis in this book. Instead of building and testing physical prototypes of the mechanism, you may use Mechanism software tool to evaluate and refine the mechanism before finalizing the design and entering the functional prototyping stage. Mechanism 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 trains, size springs and dampers, and determine interference between parts, 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 Mechanism, will help you become a more experienced and competent design engineer.

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

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

1.2 What is Mechanism?

Mechanism is a computer software tool that supports engineers to analyze and design mechanisms. Mechanism is a module of the PTC Creo product family developed by Parametric Technology Corporation. This software supports users to create virtual mechanisms that answer general questions in product design such as those described next. An internal combustion engine shown in Figures 1-1 and 1-2 is used to illustrate some of the typical questions in mechanism design.

1. Will the components of the mechanism collide 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 fast will the components move; e.g., the longitudinal motion of the piston?
4. 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.

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 Mechanism may help you answer these common questions accurately and realistically, as long as the motion model is properly defined. The capabilities available in Mechanism 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 design 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.

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 properly. At the assembly level, design parameterization involves defining assembly mates and relating dimensions across parts. When a solid model is fully parameterized, a change in dimension value can be automatically propagated 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 engine example, a change in the bore diameter of the engine case will alter not only the geometry of the case itself, but 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.
Lesson 1: Introduction to Mechanism

1.3 Mechanism 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 (or bodies), which are connected by connections (or joints), such as a pin 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 and design purposes. 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 solve in order to answer general questions regarding mechanism analysis and design. They are kinematics and dynamics.

Kinematics is the study of motion without regard for the forces that cause the motion. A kinematic mechanism must be driven by a servo motor (or driver) so that the position, velocity, and acceleration of each link in 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.

Dynamics 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 covered in Sophomore Dynamics—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,

\[ \sum F = p(t) - kx - cx = m\ddot{x} \]  

(1.1)

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

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,

\[ \sum M = -mg\ell \sin \theta = J\ddot{\theta} = mt^2\ddot{\theta} \]  

(1.2)

where \( M \) is the external moment (or torque), \( J \) 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.

Dynamics of a multi-rigid body system, such as those illustrated in Figure 1-4, 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 loads at the joint connections hold the bodies together.

Note that in *Mechanism*, you may create a kinematic analysis model, for example, using a servo motor to drive the mechanism before carrying out a dynamic analysis. In this case, position, velocity, and acceleration results may be similar to those of dynamic analysis; however, the inertia of the bodies will be taken into account for dynamic analysis; therefore, reaction forces will be calculated between bodies.
1.4 Mechanism Capabilities

Overall Process

The overall process of using Mechanism for designing or analyzing a mechanism consists of three main steps: model creation, analysis, and result visualization, as illustrated in Figure 1-8. Key entities that constitute a motion model include a ground body that is always fixed, bodies that are movable, connections (or joints) that connect bodies, servo motors (drivers) that drive the mechanism for kinematic analysis, loads, and the initial conditions. More details about these entities are discussed later in this lesson.

The analysis capabilities in Mechanism include position (initial assembly), static (equilibrium configuration), motion (kinematic and dynamic), and force balance (to retain the system in a prescribed configuration). For example, the position analysis brings bodies closer within a prescribed tolerance at each connection to create an initial assembled configuration of the mechanism. More about the analysis capabilities in Mechanism 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 the time domain. You may also query results at specific locations for a given time frame. Furthermore, you may ask for a report on results that you specified, such as the acceleration of a moving body in the time domain.

Operation Mode

Mechanism is embedded into PTC Creo. It is indeed an integrated module of Creo, and transition from Creo to Mechanism is seamless. All the solid models, placement constraints, etc. defined in Creo are automatically carried over into Mechanism. Mechanism can be accessed through menus and windows inside Creo. The same assembly is used in both Creo and Mechanism.

Body geometry is essential for mass property computations in motion analysis. In Mechanism, all mass properties are ready for use. In addition, the detailed part geometry for interference checking is also available.

User Interface

User interface of the Mechanism is identical to that of Creo, as shown in Figure 1-9. PTC Creo users should find it is straightforward to maneuver in Mechanism.

The Creo window consists of the following elements: ribbon, File menu, toolbars, navigation area, Graphics window, Creo browser, and status bar.
Lesson 1: Introduction to Mechanism

The ribbon contains the command buttons organized within a set of tabs. On each tab, the related buttons are grouped. Each button on the ribbon consists of an icon and a label. For example, under the Mechanism tab, Mechanism Analysis, Playback, and Measures buttons are grouped under Analysis. The buttons in Mechanism and their functions are summarized in Table 1-1.

The File (pull-down) menu in the upper-left corner of the Creo window contains commands, such as Open, Save, and Close, for managing files, models, and preparing models for distribution.

The Quick Access toolbar is available regardless of which tab is selected on the ribbon. By default it is located at the top of the Creo window. It provides quick access to frequently used buttons, such as buttons for opening and saving files, undo, redo, regenerate, close windows, switch windows, and so on. In addition, you can customize the Quick Access toolbar to include other frequently used buttons and cascading lists from the ribbon.

The Graphics toolbar is embedded at the top of the graphics window. The buttons on the toolbar control the display of graphics. You can hide or display the buttons on the toolbar. You can change the location of the toolbar by right-clicking and choosing a location from the shortcut menu.

The navigator includes the Model Tree, Layer Tree, Detail Tree, Folder browser, and Favorites. The Model Tree tab on the status bar (located at the lower left corner of the Creo window) controls the display of the navigator. When the Mechanism button is clicked under the Applications tab, the model tree is split into two; entities of parts and assemblies are listed in the upper tree, and entities of Mechanism are listed in the lower tree, as shown in Figure 1-9.
Table 1-1  The Major Buttons and Groups of *Mechanism*

<table>
<thead>
<tr>
<th>Group</th>
<th>Symbol</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis</td>
<td></td>
<td>Mechanism Analysis</td>
<td>Define and run an analysis.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Playback</td>
<td>Play back the results of your analysis run. You can also save or export</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Measures</td>
<td>Create measures, and select measures and result sets to display. You can</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drag Components</td>
<td>Drag components in an assembly.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gears</td>
<td>Create gear pairs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cams</td>
<td>Create a cam-follower connection.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3D Contacts</td>
<td>Define a 3D contact between two parts in different bodies.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Belts</td>
<td>Create a belt and pulley system.</td>
</tr>
<tr>
<td>Inserts</td>
<td></td>
<td>Servo Motors</td>
<td>Define a servo motor (driver).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Force Motors</td>
<td>Define a new force motor.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Forces/Torques</td>
<td>Define a force or a torque.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bushing Loads</td>
<td>Create a Bushing Load feature to simulate movement between two bodies.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Springs</td>
<td>Define a new spring.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dampers</td>
<td>Define a new damper.</td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td>Mass Properties</td>
<td>Specify mass properties for a part or specify density for an assembly.</td>
</tr>
<tr>
<td>and Conditions</td>
<td></td>
<td>Gravity</td>
<td>Define gravity.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Initial Conditions</td>
<td>Specify initial position snapshots, and define the velocity initial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Termination</td>
<td>Specify conditions to terminate the motion analysis.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Conditions</td>
<td>conditions for a point, motion axis or body.</td>
</tr>
</tbody>
</table>

The *Graphics* window, to the right of the navigator, displays the motion model with which you are working.

The *Creo* browser provides access to internal and external Web sites. Clicking the browser button on the status bar controls the display of the browser.

The status bar is located at the bottom of the *Creo* window. When applicable, the status bar displays the controls, such as the display of the navigation area, and one-line messages related to work in a window. Right-click in the message area and then click *Message Log* to review past messages.
Defining Motion (or Mechanism) Entities

The basic entities of a motion (or simulation) model created in Mechanism consist of ground, bodies, connections, initial conditions, drivers, and loads. Each of the basic entities will be briefly introduced. More details can be found in later lessons.

Ground Body

A ground (or ground body) represents a fixed location in space. The root assembly is always fixed; therefore, it becomes ground body (or part of the ground body). Also, the datum coordinate system of the root assembly is assigned as the WCS (World Coordinate System) by default. All datum features and parts fixed to the root assembly are part of the ground body.

Bodies

A body represents a single rigid component (or link) that moves relative to the other body (or bodies in some cases). A body may consist of several Creo parts fully constrained using placement constraints. A body must contain a local coordinate system (LCS), body points (created as datum points), and mass properties. Note that body points are created for defining connections, force applications, etc.

A spatial body consists of three translational and three rotational degrees of freedom (dof's). That is, a rigid 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 LCS to WCS, which is fixed to the ground body.

In Mechanism, the LCS is assigned automatically, usually, to the default datum coordinate system of the body (either part or assembly), and the mass properties are calculated using part geometry and material properties referring to the LCS. Datum axes and points are essential in creating a motion model since they are employed for defining connections and the location of external load application.

Connections

A connection in Mechanism can be a joint, cam, or gear that connects two bodies. Typical joints include pin, slider, bearing, ball, cylinder, etc. The connection will constrain the relative motion between bodies. Each independent movement permitted by a connection is called a degree of freedom (dof). The degrees of freedom that a connection allows can be translation and rotation along three perpendicular axes, as shown in Figure 1-10.

There are two types of options available for assembling parts, User Defined (joint) and Automatic (placement). The User Defined type consists of commonly employed joints for defining motion models, such as rigid, pin, ball, bearing, planar, cylinder joints, etc., which directly correspond to connections implemented in the physical mechanism. The Automatic constraints are placement constraints we employed for assembling parts, such as coincident. Both types eliminate designated degrees of freedom between components. For example, a ball joint is created simply by coinciding two datum points in their respective bodies, allowing all three rotational dof's. Some of the joints are directly equivalent to placement constraint. For example, mating two planar faces is equivalent to defining a planar joint.
Lesson 1: Introduction to Mechanism

However, instead of completely fixing all the movements, certain dof’s (translations and/or rotations) are left to allow the desired movement for a motion model.

The connections produce equal and opposite reactions (forces and/or torques) on the bodies connected. The symbol of a given joint tells the translational and/or rotational dof that the joint allows in regard to movement. Understanding the basic four symbols shown in Figure 1-11 will enable you to read any existing joints in motion models. More details about joint types available in Mechanism will be discussed in later lessons. A complete list of joints available in Mechanism can be found in Appendix A.

![Basic Joint Symbols](image)

(a) Rotation Only; e.g., Pin Joint
(b) Translation Only; e.g., Slider Joint
(c) Translation and Rotation; e.g., Cylinder Joint
(d) No Axes—Any Rotation; e.g., Ball Joint

**Figure 1-11 Basic Joint Symbols**

*Degrees of Freedom*

As mentioned earlier, an unconstrained body in space has six degrees of freedom, three translational and three rotational. When joints are added to connect bodies, constraints are imposed to restrict the relative motion between them. For example, a pin joint allows one rotational motion between bodies. As defined in the engine example shown in Figure 1-12, joint Pin1 restricts movement on five dof’s so that only one rotational motion is allowed between the propeller assembly and the ground body (case.asm). Since the engine case is a ground body, the propeller assembly will rotate along the axis of the pin joint, as illustrated in the symbol shown in Figure 1-12. Therefore, there is only one degree of freedom left for the propeller assembly.

![A Typical Motion Model in Exploded View](image)

**Figure 1-12 A Typical Motion Model in Exploded View**
For a given motion model, you can determine its number of degrees of freedom using the Gruebler’s count. *Mechanism* uses the following equation to calculate the Gruebler’s count:

\[ D = 6M - N - O \]  

(1.3)

where \( D \) is the Gruebler’s count representing the overall 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 joints, and \( O \) is the number of motion drivers defined in the system.

The single-piston engine shown in Figure 1-12 consists of three bodies (excluding the ground body), two pin joints, one slider joint, and one bearing joint. A pin or slider joint removes five degrees of freedom, and a bearing joint removes two dof’s. In addition, a motion driver is added to the rotational dof of the joint \( Pin1 \). Therefore, according to Eq. 1.3, the Gruebler’s count for the engine example is

\[ D = 6 \times 3 - (3 \times 5 + 1 \times 2) - 1 \times 1 = 0 \]

In general, a valid motion model should have a Gruebler’s count 0. However, in creating motion models, some joints remove redundant dof’s. For example, two hinges, modeled using two pin joints, support a door. The second pin joint adds five redundant dof’s. The Gruebler’s count becomes

\[ D = 6 \times 1 - 2 \times 5 = -4 \]

For kinematic analysis, the Gruebler’s count must be equal to or less than 0. The 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.

If the Gruebler’s count is less than zero, the solver will automatically remove redundancies. In this engine example, if the bearing joint between the connecting rod and the crank shaft is replaced by a pin joint, the Gruebler’s count becomes

\[ D = 6 \times 3 - 4 \times 5 - 1 \times 1 = -3 \]

To get the Gruebler’s count to zero, it is often possible to replace joints that remove a large number of constraints with joints that remove a smaller number of constraints and still restrict the mechanism motion in the same way. *Mechanism* detects the redundancies and ignores redundant dof’s in all analyses. 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 joints that will impose non-redundant constraints and still allow for the intended motion. When this is not feasible, cautions must be exercised in checking results of dynamic simulation, especially for reaction forces. Examples included in this book should give you some ideas in choosing proper joints.

**Loads**

Loads are used to drive a mechanism. Physically, loads are produced by motors, springs, dampers, gravity, etc. A load entity in *Mechanism* can be a force or torque. The force and torque are represented by an arrow and double-arrow symbols, respectively, as shown in Figures 1-13 and 1-14. Note that a load can be applied to a body, a point in a body, or between two points in different bodies.
Drivers or Servo Motors

Drivers or servo motors are used to impose a particular motion on a mechanism. Servo motors cause a specific type of motion to occur between two bodies in a single degree of freedom. Servo motors specify position, velocity, or acceleration as a function of time, and can control either translational or rotational motion. The driver symbol is shown in Figure 1-15.

Note that a driver must be defined along a movable axis of the joint you select. Otherwise, no motion will occur. When properly defined, drivers will account for the remaining dof's of the mechanism calculated using Eq. 1.3.

An example of a typical motion model created using Mechanism is shown in Figure 1-12. In this engine example, twenty six Creo parts are grouped into four bodies. In addition, four joints plus a driver are defined for a kinematic analysis.

Types of Mechanism Analyses

There are five analysis options supported in Mechanism: position, static, kinematic, dynamic, and force balance.

The position (or assembly analysis) that brings the mechanism together, as illustrated in Figure 1-16, is performed before any other type of analysis. The assembly analysis determines an initial configuration of the mechanism based on the body geometry, joints, and initial conditions of bodies. The points, axes, or planes chosen for defining joints will be brought within a small prescribed tolerance.

Static analysis is used to find the rest position (equilibrium condition) of a mechanism, in which none of the bodies are moving. Static analysis is related to mechanical advantage—for example, how much load can be resisted by a driving motor. A simple example of the static analysis is shown in Figure 1-17.
Motion analysis consists of both kinematic and dynamic analyses. 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 servo motor for a kinematic analysis, where the position, velocity, and acceleration of each link of the mechanism can be analyzed at any given time. For example, a servo motor drives a mechanism at a constant angular velocity in Figure 1-18.

![Figure 1-17 Static Analysis](image)

Dynamic analysis is used to study the mechanism motion in response to loads, as illustrated in Figure 1-19. This is the most complicated and common, but usually more time-consuming analysis.

Force balance calculates the required force and torque to retain the system in a prescribed configuration.

**Viewing Results**

In *Mechanism*, 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, as shown in Figure 1-20.
You may choose a joint or a point to generate result graphs; for example, the graph in Figure 1-21 shows the angular position of a simple pendulum example (please see Lesson 4 for more details). These graphs give you a quantitative understanding on the behavior of the mechanism. You may also pick a point on the graph to query the results of your interest at a specific time frame. In addition, you may ask Mechanism for a report that includes a complete set of results output in the form of numerical data.

In addition to the capabilities discussed above, Mechanism allows you to check interference between bodies during motion (more to be discussed in Lesson 5). Furthermore, the reaction forces calculated can be used to support structural analysis using, for example, Creo Simulate Structure, a p-version finite element analysis module of PTC Creo.

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.sdepub.com, unzip them, and locate the engine assembly model (lesson1.asm) under Lesson 1 folder. Copy Lesson 1 to your folder (chosen for working directory).

Start PTC Creo, set working directory to Lesson 1, and open the assembly model: lesson1.asm. You should see an assembled engine model similar to that of Figure 1-1.

To enter Mechanism, simply click the Applications tab on top of the Graphics window, and choose Mechanism button.

You should see Mechanism window layout similar to that of Figure 1-9. To replay results click the Playback button on top. The Playbacks dialog box (Figure 1-22) appears. In the Playbacks dialog box, click the Open button, and select the previously saved playback file AnalysisDefinition1.pbk (this file is included in the Lesson 1 folder). Click the Play current result set button at the top left corner. The Animate dialog box (Figure 1-23) opens. Click the Play button to play the motion of the engine. You should see the motion animation similar to that of Figure 1-20.

To close the model, choose File > Close. From the Home tab, click the Erase Not Displayed button in the Data group. The Erase Not Displayed dialog box opens (Figure 1-24). Click OK to erase all parts and assemblies temporarily stored in the memory.
### 1.6 Motion Examples

Numerous motion examples will be introduced in this book to illustrate step-by-step details of modeling, analysis, and result visualization capabilities in *Mechanism*. We will start with a simple ball throwing example. This example will give you a quick start and a quick run-through on *Mechanism*. *Lessons 3 through 8* focus on modeling and analysis of basic mechanisms. In these lessons, you will learn various joint types, including pin, slider, rigid, etc.; connections, including springs, gears, cam-followers; drivers and forces; various analysis types; and measures and results. *Lessons 9 and 10* are application lessons, in which real-world mechanisms will be introduced to show you how to apply what you learn to more complicated applications. All examples and main topics to be discussed in each lesson are summarized in the Table 1-2.

Note that example files have been prepared for you to go through all the lessons. In addition to *PTC Creo* parts and assemblies, each lesson folder contains complete motion models as well as simulation result files. You may want to open the motion models and review the simulation results; e.g., play motion animations, to become more familiar with the simulations before going through the lessons.

#### Table 1-2 Summary of Lessons and Motion Examples in this Book

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Title</th>
<th>Example</th>
<th>Problem Type</th>
<th>Things to Learn</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Single-Piston Engine</td>
<td><img src="image1.png" alt="Image" /></td>
<td>Kinematics</td>
<td>1. General introduction</td>
</tr>
</tbody>
</table>
| 2      | Ball Throwing Example  | ![Image](image2.png) | Particle Dynamics | 1. This lesson offers a quick run-through on general modeling and analysis capabilities in *Mechanism*.  
2. You will learn the general process of using *Mechanism* to construct a motion model, run analysis, and visualize the motion analysis results.  
3. Simulation results are verified using analytical equations of motion. |
| 3      | Spring-Mass System     | ![Image](image3.png) | Particle Dynamics | 1. This is a classical spring-mass system example you learned in *Dynamics*.  
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. Simulation results are verified using analytical equations of motion. |
| 4      | A Simple Pendulum      | ![Image](image4.png) | Particle Dynamics | 1. This lesson goes more in depth about creating joints in *Mechanism*. Pin and rigid joints will be introduced.  
2. Simulation results are verified using analytical equations of motion. |
Table 1-2 Summary of Lessons and Examples in this Book (Cont’d)

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Title</th>
<th>Example</th>
<th>Problem Type</th>
<th>Things to Learn</th>
</tr>
</thead>
</table>
| 5      | A Slider Crank Mechanism                    | ![Slider Crank Mech](image) | Multibody Kinematic and Dynamic Analyses | 1. This lesson uses a slider-crank mechanism to discuss more joint types as well as conduct position (initial assembly), kinematic, and dynamic analyses.  
2. In addition to joints, you will learn to create drivers for motion analysis.  
3. The interference checking capability will be discussed.  
4. Simulation results are verified using analytical equations of motion. |
| 6      | A Compound Spur Gear Train                  | ![Compound Spur Gear](image) | Gear Train Analysis                | 1. This lesson focuses on simulating motion of a spur gear train system.  
2. You will learn how to use Mechanism to create a gear connection, analyze the gear train, and define measures for gears.  
3. Simulation results are verified using analytical equations. |
| 7      | Planetary Gear Train Systems                | ![Planetary Gear](image) | Planetary Gear Train Analysis      | 1. This lesson is similar to Lesson 6 but focuses on planetary gear trains.  
2. Both single gear and multiple gears systems will be discussed.  
3. Some simulation results are found incorrect using analytical equations. |
| 8      | Cam and Follower                            | ![Cam and Follower](image) | Multibody Kinematic Analysis      | 1. This lesson discusses cam and followers.  
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 created to simulate the motion of the system as well as assess the engineering design of the system. |
| 9      | Assistive Device for Wheelchair Soccer Game | ![Assistive Device](image)  | Multibody Dynamic Analysis        | 1. This is an application lesson. This lesson shows you how to assemble and simulate motion of an assistive device for playing wheelchair soccer game.  
2. Numerous joints, spring, and force will be created for the system.  
3. Measures will be defined to assess the design of the system. |
Table 1-2  Summary of Lessons and Examples in this Book (Cont’d)

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Title</th>
<th>Example</th>
<th>Problem Type</th>
<th>Things to Learn</th>
</tr>
</thead>
</table>
| 10     | Kinematic Analysis for Racecar Suspension  |         | Multibody Kinematic Analysis | 1. This is the second and the last application lesson of the book. A quarter of a racecar suspension will be employed for kinematic analyses.  
2. A road profile will be modeled by using a cam of special profile. The cam will be connected to the tire using a cam-follower connection.  
3. Various measures, including the camber angle, will be introduced to assess the design of the suspension system. |