Most of the fast airplanes have retractable landing gear. You must extend it in the ground scenes and retract in the pictures of the flying airplane. Of course, it is possible to move and rotate each of the landing gear parts separately. However, I assure you that such an exercise will become very tedious, when you repeat it several times. The better idea is to use the rich set of Blender animation tools. In this article, I will show you my solution for controlling the movable parts of an aircraft model.

I will explain this on the example of a tail wheel assembly (from the model in the picture above). Let's take a look at this mechanism (Figure 1). The wheel is pivoted on the fork, which is attached at the end of the swing arm. There is also a shock absorber and an actuator, which, when moved, pulls forward the shock absorber, rotating the swing arm and retracting the tail wheel.

We can easily recreate the base mechanism using an armature. It will consist of 3 bones (Figure 2):
Unfortunately, when the modeled elements have to perform spatial movements, the bones have tendency to rotate around their longitudinal axes. This can lead to surprising results. For example, I tried to use similar “three-bone” setup for the tail wheel doors retraction mechanism (Figure 3):

![Figure 3 Problems with the doors intermediate positions](image)

You still could master such an “unruly” armature, but, frankly, I did not want to spend much time on such trials. During my attempts to cope with the bone torsions I learned more about object constraints. Then I quickly came to the conclusion that it is possible to extend and retract the landing gear without any bones, by a single shift of an auxiliary handle. You just need to reverse the cause and effect sequence. The swing arm will not be dragged by the shock absorber, just the absorber will move, because it will be pushed by the rotating swing arm. When you do it right, the external observer will not notice the real relationships between various parts of this mechanism. Using such a solution you can model entire assembly using a few constraints. The armatures with their Pose Mode are no longer needed.

I started by creating an auxiliary handle object, named X.500.Handle (on a layer which is not rendered). Then I used a Locked Track constraint to direct the swing arm toward this handle (Figure 4):

![Figure 4 Using a Locked Track constraint to retract the tail wheel](image)

In such a setup like in Figure 4, when I move the handle upward, the tail wheel retracts. To extend it, I have to move the handle downward. In general, you can use the Locked Track constraints to create “pivot joints” in your mechanisms. So far, it is nothing extraordinary, isn’t it?
Let's start to enhance this simple scheme. For example, you can use a Limit Location constraint to restrict the handle movement to a "reasonable" range (i.e. a range that corresponds to realistic positions of the controlled mechanism). Figure 5 shows such a restriction imposed on the handle from Figure 4:

![Figure 5 Restrictions imposed on handle location](image)

In this way, you do not have to worry whether you set the handle in the appropriate location. Now it is forced by the constraint.

To include into this virtual mechanism the shock absorber and the actuator, you have to choose carefully the origins of these objects (Figure 6):

![Figure 6 Origins of the tail wheel assembly elements](image)

In the mechanism like this, the actuator pushrod and the shock absorber piston should have their origins in the same place. The logical hierarchy (the Parent relationships) of these objects is also important. Assign the piston to the pushrod, the shock strut to the connector object, and the connector to the swing arm (see Figure 6).

Use another Locked Track constraint to direct the connector (and its "child" — the shock strut) objects toward the pushrod origin (Figure 7):

![Figure 7 Constraining the connector object](image)

Assign two other Locked Track constraints to both shock absorber parts, to direct their Z-axes toward the parent of the opposite element (Figure 8):

![Figure 8 Constraining the shock absorbers parts](image)

Such a constrained shock absorber deflects and extends, following movements of the handle. However, we still have to force the actuator pushrod to perform the linear movement. How to obtain such an effect?
Retracting Landing Gear

For the actuator pushrod, we will use an “all-purpose gear”: the *Transformation* constraint (Figure 9):

- **Y range**: -3.5 unit

The tracked object: X.500.Handle

Every change of the local Z coordinate of the handle causes a proportional change of the Y coordinate of the pushrod.

The variation of the corresponding pushrod coordinates and directions above are expressed in the local coordinate system of the corresponding objects.

All these coordinates and directions above are expressed in the local coordinate system of the corresponding objects.

In the result, a single handle movement drives the tail wheel mechanism (Figure 10):

- **Extended tail wheel**

Of course, if you want to create an animation sequence, you can assign to this handle object a movement curve (*F-Curve*). It is simpler than dealing with the armature bones!
There is still one detail missing: the shock absorber deflection that occurs under various loads (the static weight of the airplane standing on the ground, or the dynamic forces during landing). To reproduce this effect, I use a “child” handle (Figure 11):

I shaped this child object as a “donut” around the original handle — just to grab it easily, when needed. Initially its origin is in the same place as the origin of its parent — the main handle object (X.500.Handle). This child handle has a Limit Location constraint, which allows it to move within the range of 2 Blender units from its parent. (Note that you still have to enter the corresponding Maximum and Minimum values as you can see them in the Object properties panel — see Figure 11).

When the child handle is set, use it in the Locked Track constraints of the swing arm and the shock absorber objects (Figure 12):

The swing arm object constraint

- Target: X.501.Handle
- Vertex Group: X.501.Handle
- To: X Z
- Lock: X Y Z
- Influence: 1.000

Figure 12 The effect of landing gear deflection

To close and open the tail wheel doors, assign to their hinges Transformation constraints, which convert handle location into rotations (Figure 13):

Figure 13 Closing the tail wheel doors
The last element we have to animate is the door retraction mechanism (Figure 14):

It consists of a central pivot, which connects two drag braces. The end of each brace is connected to the barrel on the door fitting. I used Lock To constraints to control the pivot and barrels directions.

The braces have to use the Stretch To constraints, because of linear rotation of the doors (Figure 15):

Figure 16 shows the ultimate result: retraction of the complete tail wheel assembly. I drew on it the handles, because they are not rendered:
You can use similar solution for the main wheel. In the case of the P-40 it was simultaneously rotating around two axes (Figure 17):

![Figure 17 Main wheel constraints](image1)

I applied appropriate *Transformation* constraints to rotate the leg and its pivot (the latter part is hidden inside the leg and its fittings — see Figure 17).

Another interesting part of this mechanism is the scissor link (Figure 18). Its movement is caused by the shock absorber deflection. The easiest way to animate such pivoted struts is the armature.

![Figure 18 Scissor link constraints](image2)

The key element of this scissor link assembly is the central bolt. Its position is determined by two *Limit Distance* constraints, which keep it on fixed distance from the upper and lower bolts. The two struts have their origins at the corresponding bolts (the upper or lower one). They are stretched toward the central bolt using the *Stretch To* constraints. I used this constraint here instead of the *Locked Track*, to hide minor gaps between their ends and the bolt (I cannot explain why they occur).

Figure 19 shows the retraction of the complete main wheel.

![Figure 19 Retraction of the P-40 main wheels](image3)
Except the undercarriage, there are also other movable parts which are worth to “mechanize”: the rudder, elevator, ailerons, wing flaps... For each of them I created a handle. Finally, I ended up with dozens of different handles, “orbiting” around the model (Figure 20). To manage them, I created a universal add-on. It scans the current scene for handle objects and creates their panels in the Properties window. To do this “magic”, just append to the names of your handles the “.Handle” suffix, and assign to them Limit Location constraints. Using this tool, I can quickly pose the airplanes in such a scene like the one shown in Figure 21:

The model presented in this article is available at [http://airplanes3d.net/model-p40_e.xml](http://airplanes3d.net/model-p40_e.xml)
The Handle Panel add-on and its description is available at [http://airplanes3d.net/tutorials-020_e.xml](http://airplanes3d.net/tutorials-020_e.xml)

This article is based on a more detailed description, presented in volume 4 of the “Virtual Airplane” book. (See [www.airplanes3d.net](http://www.airplanes3d.net))