SOLIDWORKS

The Ultimate Guide to

SKELETON MODELLING

Yusuf Birader

The Ultimate Guide to Skeleton Modelling in SolidWorks

Foreword

When I first started learning SolidWorks, I was pleasantly surprised by how easy it was to get started. Within a few weeks, I knew my extrudes from revolves and lofts from sweeps. But there was a problem. My models were brittle and hard to change- modifications made to one part would ripple through the entire model, causing it to break and make my SolidWorks UI light up like a traffic light. Instead of spending time iterating on my product designs, I would spend tedious hour after hour fixing my broken models. It got to a point where I was scared to make any change! The reason was that while I knew how to *use* SolidWorks, I didn't know how to *design* with it. So I set off to understand the essence of design, what makes a good design, and how particular modelling strategies help to develop a good design.

It's what led me to discover *skeleton modelling*. It's an approach used widely by industry veterans that combines the high-level, iterative advantages of top-down design with the robustness and manufacturability of bottom-up design. With skeleton modelling, my designs became quick to model and even easier to change.

It was at this point that I thought- if only there had been a guide I could have followed that would have fast-tracked me from beginner to effortlessly designing like a pro.

I hope you find it useful.

Introduction

When beginners first start using SolidWorks, their only priority is to get to grips with the ins and outs of 3D modelling.

In practice, this usually means brute forcing their way through designs. It may eventually lead to a functioning assembly, but the model will be brittle, hard to understand, and even harder to change.

On the other hand, those who have reached SolidWorks enlightenment are able to produce robust models that are a joy to work with.

This difference is what separates the novices from the professionals.

On my journey to reach this promised land, I've come across many who cite the virtues of "good design". I found myself nodding in agreement- good design is indeed the goal.

The problem is that I didn't actually know what good design was when it came to 3D modelling nor how to practically achieve it.

If you're reading this and are anything like me, you probably have the same questions.

So, in this guide, I'm going to go through *why* we need design, the *principles* of good design in CAD, and *how* we can develop practical modelling strategies that use these principles to create robust, well-designed models.

Although I describe these concepts in the context of SolidWorks, they generalise to any parametric CAD program.

Why Design?

In the simplest case, each model in SolidWorks starts life with a sketch. These sketches form the basis of features. These features are then composed to form the solid model.

In an abstract sense, the model is essentially a graph of entities, where each entity depends on zero or more other entities.

The topology of this graph of entities defines the *design* of the model. In other words, *the design of a model is the arrangement of entities and the relationships between them.*

Consider the case where we're asked to create a model for a new product design. And assume that the requirements we're given for the model are complete and correct- we know what all the dimensions and constraints should be and we're reassured by the client that once the model is drawn, it will never need to be changed.

In this case, the design of the model does not matter. We can put the sketches and features in any which way so long as the requirements of the design are met. No one topology of the graph is better than another since the arrangement of the entities will never need to be changed.

But the problem is that this *never ever happens* in real life. Things *always* change. Parts are swapped, dimensions are adjusted, and the form iterated.

The model must not only suffice today but be amenable to change tomorrow.

This is the essence of good design.

A model is well-designed if it can be easily changed.

Specifically, the model's entities should be arranged so that the effort to change the model is proportional to the size of the change.

The benefits of this are numerous- rapid iteration speed, less technical debt, and happier engineers- all of which lead to better products.

Badly designed models have the opposite property- small changes ripple through the dependency graph, quickly resulting in unsolvable states, and your SolidWorks UI lighting up like a traffic light. Making a change in a badly