* Model design:

The V-model basically divides the development process into three sections: The decomposition on the left side of the V-model describes the transformation of requirements, which are presented as an input, into a system design. This leads to the second section of engineering in different disciplines, the domain-specific design process. The third section integrates the disciplines on the right side of the V-model during the system integration, verification and validation. The result or output of the V-model is a product. The model is supplemented by a bracket for modelling and model analysis, which begins and ends at the middle height of the two sides. Arrows between the two sides of the V-model illustrate the assurance of properties comprising verification and validation activities. Note, the assurance of properties is indicated right-to-left, which gives the impression of a retrospective process.

MIL, SIL, PIL and HIL testing come in the verification part of Model-Based Design approach after you have recognized the requirement of the component/system you are developing and they have been modeled in simulation level (e.g. Simulink platform).

First, you have to develop a model of the actual Plant (hardware) in any simulation environment, for e.g. Simulink, which captures most of the important features of the hardware system. After the Plant model is created, we will develop the Controller model and verify if the Controller can control the Plant as per the requirement. This step is called Model-in-Loop (MIL) and you are testing the Controller logic on the simulated model of the plant. If your Controller works as desired, you should record the input and output of the Controller which will be used in the later stage of verification.

Once your model is verified (i.e., MIL in the previous step is successful), the next stage is Software-in-Loop(SIL), where you generate code only from the Controller model and replace the Controller block with this code. Then run the simulation with the Controller block (which contains the C code) and the Plant, which is still the software model (similar to the first step). This step will give you an idea of whether your control logic i.e., the Controller model can be converted to code and if it is hardware implementable. You should log the input-output here and match with what you have achieved in the previous step. If you experience a huge difference in them, you may have to go back to MIL and make necessary changes and then repeat step 1 and 2. If you have a model which has been tested for SIL and the performance is acceptable you can move forward to the next step.

The next step is Processor-In-Loop (PIL) testing. In this step, we will put the Controller model onto an embedded processor/FPGA board and run a close loop simulation with the simulated Plant. So, we will replace the Controller Subsystem with a PIL block which will have the Controller code running on the FPGA board. This step will help you identify if the processor/FPGA board is capable of running the developed Control logic. If there are glitches, then go back to your code, SIL or MIL and rectify them.

Once your plant model has been verified using PIL, now you can replace the plant model with the original hardware, say lab model and run a test.

As an added step, you can have a simulated plant model in a real-time PC before interfacing with the actual hardware, for example, in Speedgoat where you will run the plant model in real-time but have proper Analog/CAN communications between the Plant and Controller which you will be using while interfacing with the plant hardware. This step is known as HIL testing and the controller is typically on a production board/controller. This will help you with identifying issues related to the communication channels for example attenuation and delay which are introduced by an analog channel and can make the controller unstable if they are not captured in the simulation. This step is typically done to test safety critical application e.g. air bag deployment.

