Abstraction Class, State Variables and Parameters

Given the system, the question, and your key frames, what class of abstraction do you expect to use for this system? In other words, how many different parts will you deal with? Will you be treating these parts as rigid bodies or as particles? Are the parts constrained in their motion, or unconstrained? Make a diagram and a table that define the state variables you will need to keep track of in order to model the dynamics of this system, as well as the relevant parameters for the system . Your table should include both a symbol for each state variable or parameter, a description of what the state variable or parameter tells you, and why it is important.

Looking at the scenarios outlined above, we can identify a number of different *phases* of motion we might want to model:

1. The time before the skateboarder goes onto the ramp. Presumably this is pretty simple – in fact, we might just want to START at the point when the skateboarder goes onto the ramp – or, even better, at the point when the skateboarder crosses the midpoint of the ramp!

2. The time when the skateboarder is on the ramp. This looks a lot more complex: the ramp can move (although it's limited by the ground to rotating through a particular angular range), and the skateboarder can move along the ramp.

3. The time after the skateboarder leaves the ramp (either by launching off the end of the ramp, or by neatly rolling off on to the ground on either the left or the right side). This is not so complex, as presumably we don't care how the ramp is moving once the skateboarder leaves the ramp. It's possible we'd want to think about the launch condition (keyframe 2 above) as distinct from the "clean" exit (keyframe 3), but given the question we want to answer, it might be enough just to know that the launch takes place.

So, even though there are multiple phases here, I propose that the one phase we really need to concentrate on is motion when the skateboarder is on the ramp.

During this phase, it's clear that the ramp and the skateboarder can EACH move – so I need at least 2 parts in this system.

The ramp's motion is clearly rotational, so I'll have to use a rigid body for that. Since it's fixed in the middle, this is a constrained rigid body.

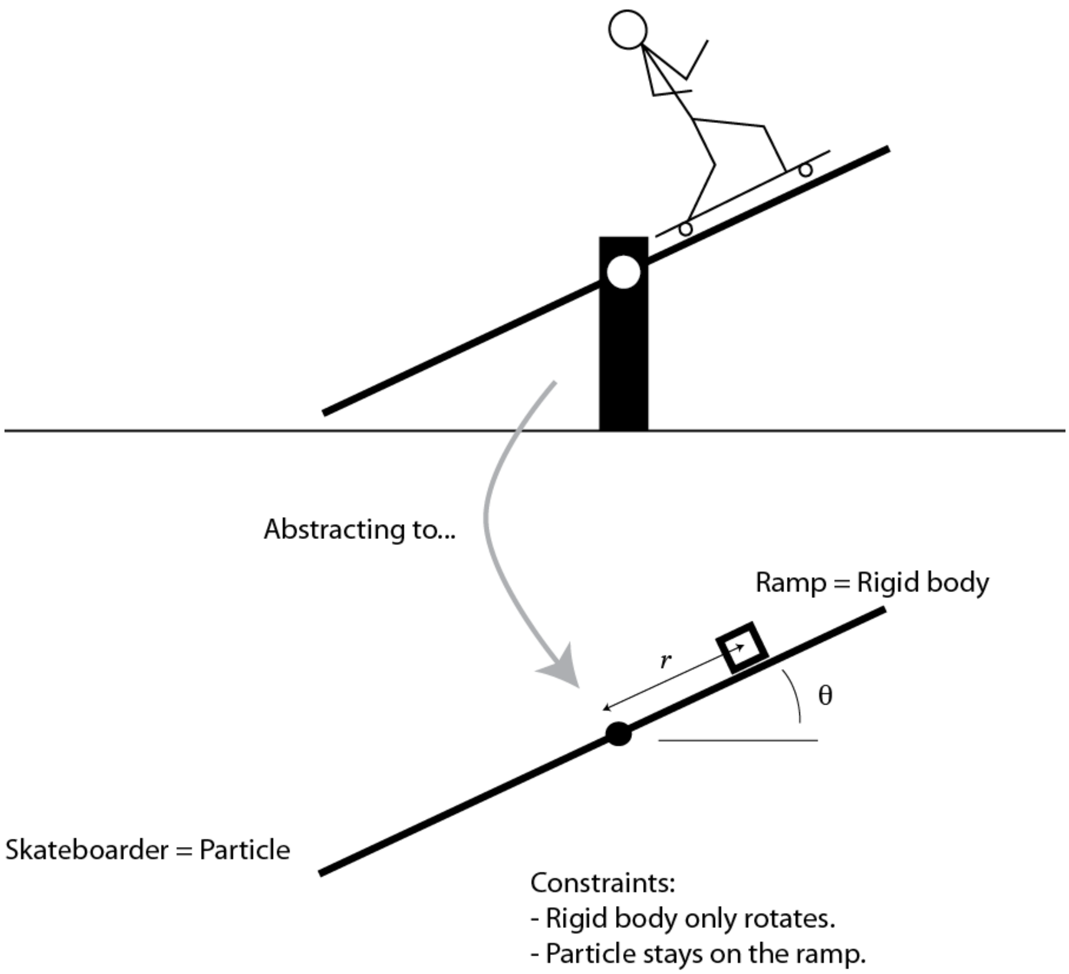
The skateboarder could be dealt with in any number of ways - from a particle (i.e., think about the skateboarder as a very compact "lump" that slides, with out without friction, along the ramp) to a rigid body that slides along the ramp and rotates, to a much more complex collection of rigid bodies (e.g., arms, legs, etc.) which are connected to each other in interesting ways (e.g., muscles and joints).

Since this is our first cut at the model, we should probably take the easiest approach that might give us some insight – i.e., treat the skateboarder as a particle.

Now, to the question of how many state variables there are: Clearly the position, and angular velocity of the ramp matter. Also, it's obvious that where the skateboarder *is* on the ramp matters, as does the skateboarder's speed. It appears that if you captured these four, you'd be in good shape.

From a parameters perspective, I would think that the mass of the skateboarder would matter, as would the mass and length of the ramp. Not sure what else...

So, my abstraction looks like this:



And my list of parameters and state variables is as follows:

State Variable/Parameter	Meaning, range, units, etc.	Why?
θ	Angular position of the ramp: runs from some positive angle to some negative angle, depending on ramp design. In radians.	Obvious by inspection.
ω	Angular velocity of the ramp: radians per second, could in principle be anything	Once the ramp is spinning, it's going to want to keep on spinning
r	Distance of rider from the pivot (positive = to the right of the pivot), in meters	The greater the distance, the greater the torque exerted
v_r	Time rate of change of r, in meters per second	Once the rider is moving, the rider wants to keep on moving.
m	Mass of the rider (kg)	Heavier riders will push down the ramp more.
L	Length of the ramp (m)	Determines both the initial angle and the length of time on the ramp, as well as the ramp's moment of inertia.
M	Mass of the ramp (in kg)	Determines the ramp's moment of inertia – bigger M means the ramp is harder to move

Now it's worth noting that the number of state variables is actually LOWER than you might guess it would be by just thinking about a particle and a rotating rigid body. For a particle, you generally need to know both its position ($\vec{r} = x\hat{\imath} + y\hat{\jmath}$), and its momentum ($\vec{p} = m(\dot{x}\hat{\imath} + \dot{y}\hat{\jmath})$, which gives you four state variables in two dimensions; for the rotating (but not translating) rigid body, you need to know its angular position (θ) and its angular momentum ($\vec{L} = I\omega\hat{k}$). However, since the sb is constrained to be on the surface of the ramp, x and y can be expressed purely in terms of how far along the ramp the sb is, and what the angle of the ramp is: $\vec{r} = r\cos\theta\hat{\imath} + r\sin\theta\hat{\jmath}$. So, we end up needing two fewer state variables than we might expect otherwise.