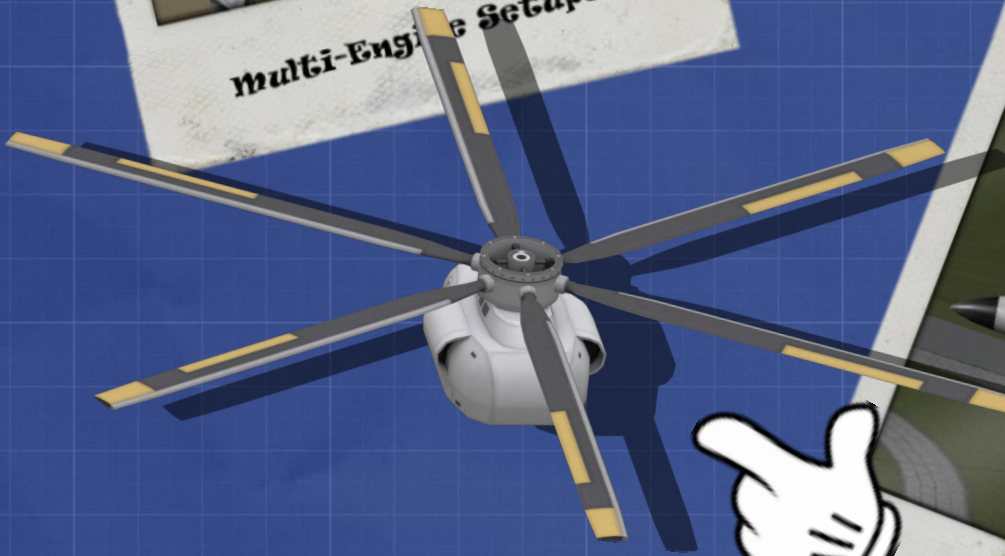


# Propellers and Spinny Things!

## Untech Engineering Manual #271 Propeller and Rotor Installation Guide



Multi-Engine Setups!



Kerbe-kepters!



**Untied  
Technologies**

## Key Binding Assignments

I have to start by mentioning my key bindings. I've used them for so long they are just second nature to me and I forget about them when explaining things to folks. So feel free to keep the assignments you like but forgive me if I mention something like "L/R Arrow" keys in some of my craft blueprints and videos and your "L/R Arrow" keys do something useless like scroll the camera.

First, I always swap the F1 and F12 key assignments. The KSP default for F1 takes a screenshot, and the default for F12 displays the aerodynamic overlay. For a game that is now distributed via Steam, this doesn't make a whole lot of sense, so I just switch them.

The default key for Steam screenshots is F12. So if I make F1-F12 switch in kerbal, my screenshots get saved to both the KSP game folder and my Steam folder. So now my F1 key in KSP displays the aerodynamic overlay. The default settings had me saving a ton of stupid screenshots in Steam every time I toggled the aero forces overlay.

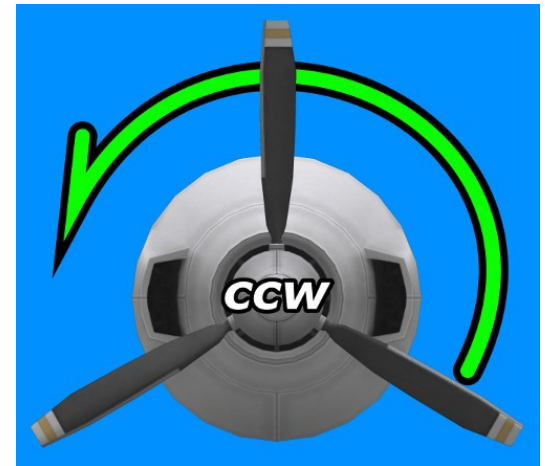
Key	Assignment
F1	Aero Forces Overlay
F12	Screenshot
L/R Arrows	Translate L/R
U/D Arrows	Translate U/D
Camera Scroll Keys	Unassigned

The other change I like to make is assigning the keyboard arrow keys to the Translate U/D and Translate L/R keys. I find this way, my hand positions are a lot more comfortable on the keyboard and I don't have to look down as much to figure out which keys my right hand should be hovering over. Since the arrow keys are assigned to camera scrolling/rotation by default, I either just delete those references or reassign them to the keypad 8, 2, 4, 6 arrow keys. I always manage the camera with the mouse so I prefer to just delete them.

## Engine Rotation

When I built my first propeller craft in KSP, I noticed the engines were setup by default with clockwise rotation. I thought, "Perfect, the default is already setup for clockwise rotation!" When I flew the craft however, I noticed that my aircraft were rolling to the right, which is exactly the opposite of what I was expecting. "So what's going on here?"

Most propeller engines in real life are described as spinning "clockwise" when viewed from the pilots seat or back of the airplane. This is the opposite of the default setting for engines and propellers in KSP. The view perspective here is the key! This can be confusing because the point of reference for "Clockwise" (CW) and "Counter-Clockwise" (CCW) rotation in KSP is stated from the perspective of viewing the engine from the front. So, if you want to mimic real life for the most common propeller rotation for your aircraft, you will want to change both the engine rotation and the blade variant to CCW.



Standard engine rotation in Real Life = CCW in KSP

## Engine Brake Default

This is a curious setup for an airplane or helicopter engine. By default, the "Brake" value of the engine is set to 100. This alone is not a big deal, but the engine brake is also assigned to the "B" key for the landing gear wheel brakes and that is just a "NO!" for me. Under no circumstances do I ever want to engage wheel brakes and stop the engine!

I have to be able to sit on the runway in my craft (airplane or helicopter) and be able to spin-up the props or rotors with the wheel brakes engaged. So I always, ALWAYS go into the key assignments for "Brakes" and delete all references to the engines.



Reassign "Engine Brake" to Key 3

If we didn't reassign the Engine Brake to key 3, the propeller might seem to spin forever at shut down. I also turn the brake value down a LOT from the default value of 100.

Setting the brake value to 1, keeps the engine from a jarring and sudden stop of the blades. Press key 3 to arrest the propeller/rotor motion and it should wind down gently to a stop over a few seconds. Just gives it a more realistic feel this way.

## Motor Size & Output

The default setting for "Motor Size & Output" is maxed-out at 100 and relates to two key items: Motor Power (think torque) and Engine Mass. In KSP

we are always looking for MOAR POWER! So not too surprisingly, the default is maxed at 100. In Figure 1 we see the 100 setting results in a power output of 150 kN (kilo-Newtons) and "extra mass" of 0.6t. It makes sense that a more powerful engine should have more pistons, parts, and so forth to achieve the greater power output.

The default RPM limit is set to the max at 460. According to the game developer notes, the limit was capped at 460 due to it being the maximum that the underlying Unity game engine physics model could handle. Sure, in a rocket design, MOAR POWER is always better! For a propeller craft however, that extra power and resulting engine torque-ness could cause your plane to literally roll over on the runway during engine startup. For your aircraft size and weight, you may need to tone that down a bit.

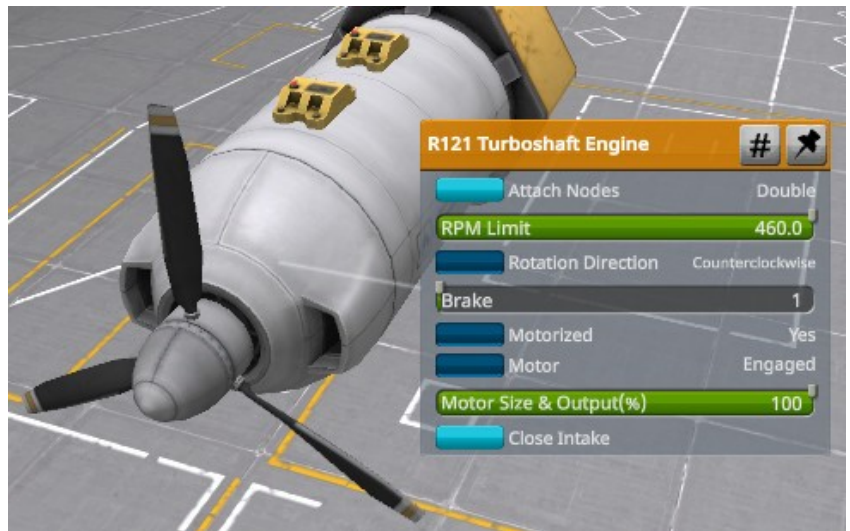
RPMs are going to be the important driving force for both your propeller and rotor driven aircraft. The extreme view might be to go as low on the engine size as possible to achieve your target RPMs. I've seen people say, "*Motor size beyond the minimum required is just wasted power.*" I disagree with that notion.

What I think you really want is to have enough engine power to maintain your target RPMs during all phases of flight. The only way you're really going to know what that power setting will be is to flight test your aircraft and adjust accordingly. How do the RPMs look during takeoff? What happens to RPM as your prop pitch changes? What about steep climbs? How might airspeed influence the RPMs of a helicopter rotor? These are all good questions. Do your flight testing; make your observations, and adjust the Motor Size & Output slider accordingly.

## P-Factor

Although I have seen folks claim in forum discussions that P-Factor is modelled in KSP, I'm not so sure I see it. I think what they are really referring to is torque-induced roll rather than P-Factor yaw. After designing and flying a few propeller craft in KSP my gut reaction is, "*No, P-Factor isn't really modelled.*" ...or at least if it is modelled, the yaw influence during flight is so minimal that I have yet to see where right/left rudder would need to be applied during high RPM high Angle of Attack (AoA) flight to counter the P-Factor yaw effect.

What people are seeing a lot in KSP (and maybe mislabeling it P-Factor?) is really over-powered (OP) engine torque causing a counter-rotation of the airframe. This introduces unwanted roll as opposed to unwanted yaw. I don't think I've seen high-torque roll described as P-Factor before, but I get what people are trying to say. I may be guilty of mislabeling it in discussions a few times myself.



Changing the Engine Brake value to 1 will bring the propeller to a gentle stop

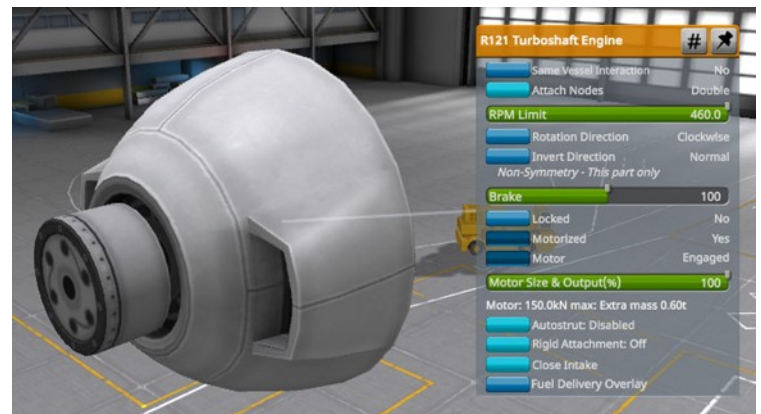
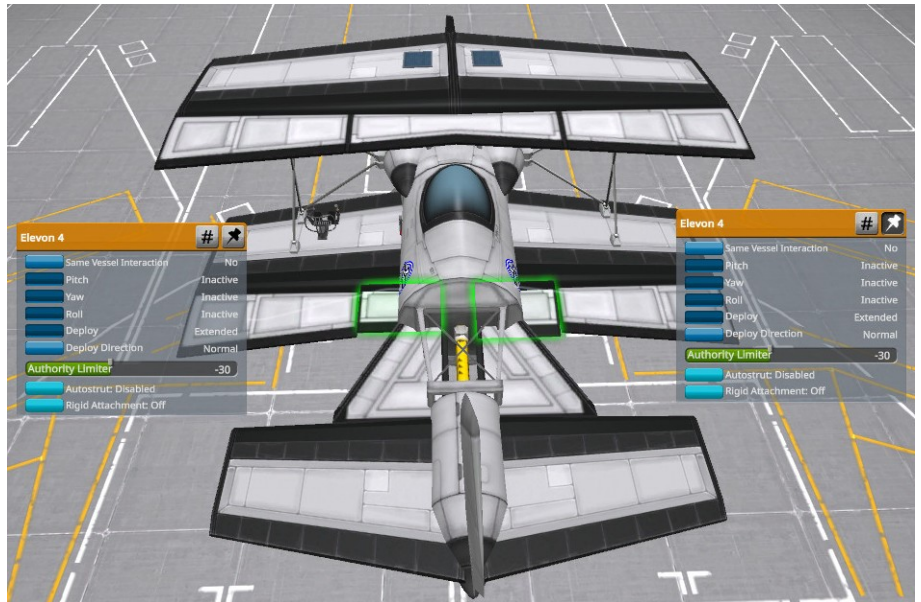


Figure 1: KSP Default Engine Settings

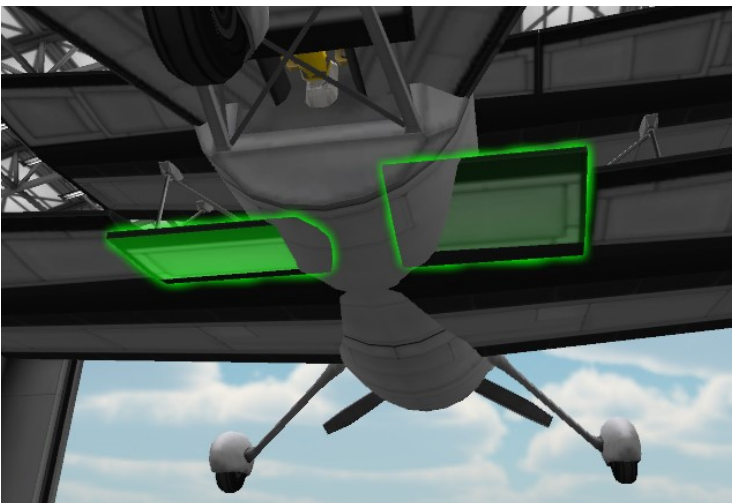
## Aileron Trim Tabs

P-Factor semantics aside, how do we counter the unwanted roll effect in our propeller aircraft designs? Two factors come into play. As mentioned previously, you will be adjusting the “Motor Size & Output” slider to reduce that torque-twisting influence on your aircraft. That alone however, is probably not going to get rid of it completely. I find that for my single engine propeller craft I also have to add aileron trim tabs. You might even have an OP stunt plane in mind with your design and the very thought of decreasing any engine power is just out of the question!



Trim tabs highlighted green and attached close to the fuselage

To counter the roll effect, aileron trim tabs can be added as additional control surfaces on each wing. Once the “Elevon 4” parts are positioned correctly with mirror symmetry, the “Remove Symmetry” button is clicked so the Authority Limiter values can be changed to positive/negative as required.



Modeling aileron trim tabs lets you save your craft’s “Trimmed for Take-off” position in the SPH so it’s ready to go when it hits the runway!

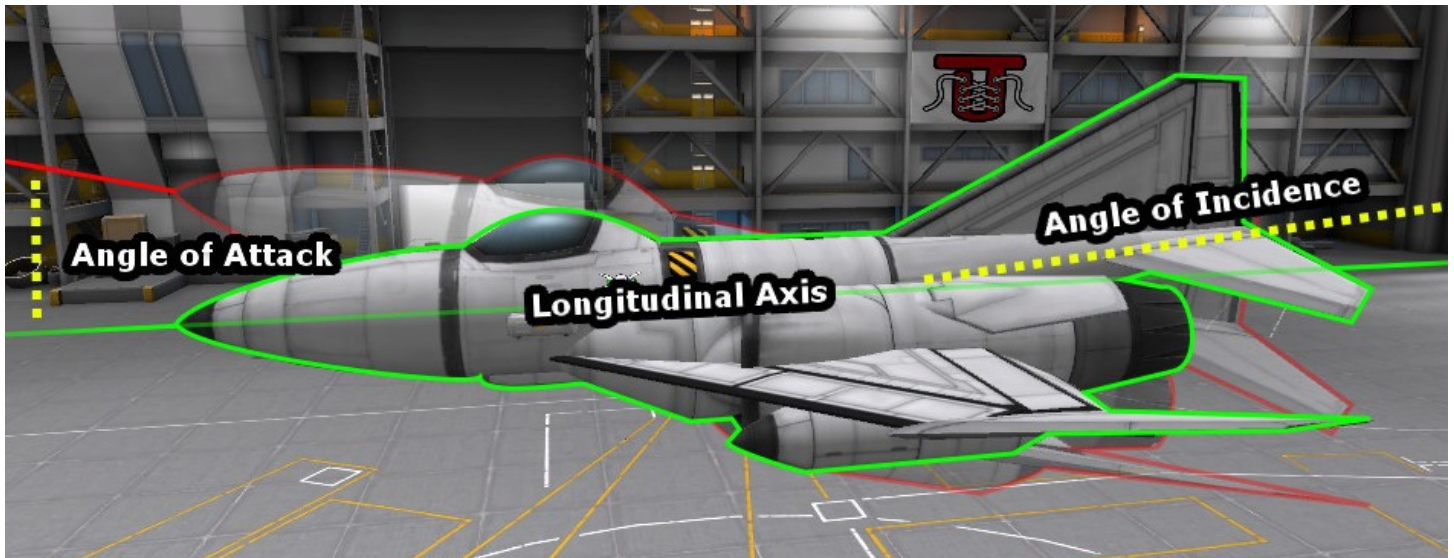
In this case, setting the Authority Limiter value to -30 on both sides caused the port trim tab to go DOWN, and the starboard tab to go UP. Where on the wings you want to install the tabs is really at your discretion as the designer. Some may prefer to have them closer to the wing tips, which would require lower values to correct for roll. For my little stunt biplane design, it seemed to make sense to add them close to the fuselage as shown. I wanted to maintain “snappy” aileron control and aesthetically I thought they looked pretty good!

I assigned the Authority Limiter values for each of these parts to the Translate L/R keys. This setup allows me to include a built-in counter for unwanted roll at design time in the SPH (Space Plane Hanger) so the biplane is trimmed for takeoff the moment it hits the runway.

During flight, if I detect a roll in one direction or another due to my changing airspeed, I can just use my left/right arrow keys to easily adjust the aileron roll trim settings to keep me flying straight and level. The standard game controls do allow for trim adjustments during flight (Alt+WASD keys) but I prefer to manage it like this so I can adjust the trim for takeoff at design time in the SPH. It also seems to be more responsive than the in-game flight trim keys.

## Elevator Trim and Angle of Incidence

I have also noticed a tendency for propeller craft to want to nose-down a bit after takeoff. The solution for unwanted pitch follows the same general principals for dealing with unwanted roll. You can add elevator trim tabs to your design and assign those to translate hotkeys if you like. In my designs so far, the unwanted pitch does not seem to be as bad as the unwanted roll. So rather than adding elevator trim tabs, I have been able to counter it with small adjustments to the elevator Angle of Incidence. I can tune the elevator, for takeoff and normal flight so the pitching down is minimal or negated.



The elevator Angle of Incidence (AoI) on this jet has been tilted forward to counter the tendency for the nose to pitch down during flight

Sometimes I'll say "Angle of Attack" when I really mean "Angle of Incidence". The concepts are similar, but the two really are different things. Angle of Attack (AoA) has to do with tilting of the Longitudinal Axis of the airplane during flight. Angle of Incidence (AoI) relates to how the control surface is attached to the airframe and tilted relative to the Longitudinal Axis of the fuselage. Above, we see a jet design where the elevator AoI was adjusted to counter the tendency of the craft to pitch down during flight.

If we make AoI adjustments to our wings and/or elevators, we can diminish or remove up/down pitching that may become evident during flight testing. Larger cargo and passenger planes may even add upward pitch to the wings to improve lift and assist during takeoff. This can also be particularly useful if your plane requires low speed takeoffs and landings with high friction landing gear (think seaplanes).

With the elevator AoI adjusted for takeoff, you might see a tendency for the nose to then pitch up a bit during normal higher speed flight. Some balancing may be required between takeoff and normal flight to minimize the up/down pitching during both phases and then just fly with SAS (T-key) turned ON to automatically smooth things out for you. Additionally, we can also use the in-game (Alt+WASD) trim keys to make the adjustments needed for straight and level flight.

## Flying with SAS

I've seen the discussions. *"If you have to fly with kerbal SAS on, your aircraft design is no good!"* I suspect the majority of these arguments are coming from people who have never actually flown a real aircraft. Adjusting the AoI of the main wing and empennage allows you to adjust for roll, pitch, and yaw in your designs and is not a particularly advanced or super-secret design concept. So I'm not entirely sure why they might think their "Trimmed for Takeoff" plane designs are so superior?

Whatever adjustments you make to trim-out your design, it's just not going to work for every phase of flight. As your airspeed and Angle of Attack (AoA) change, the control surfaces will need to be adjusted or trimmed to keep your nose straight and level. Not to mention the changing weight and balance caused by fuel being spent. Advanced aircraft have fly-by-wire and other automated systems to make these types of adjustments for the pilot automatically. Smaller and less sophisticated general aviation aircraft have no such automation and the pilot is responsible for adjusting trim during flight.

I view SAS as an auto-trim or simplistic fly-by-wire implementation. To me, it makes our little kerbal contraptions actually feel more like flying a real plane. It also helps to smooth out some airframe flutter that you might notice when your thrust is perhaps a bit too OP for your design. I know, I know, *"Who would ever build an OP jet in KSP?"* I don't pay any attention to those who criticize SAS usage.

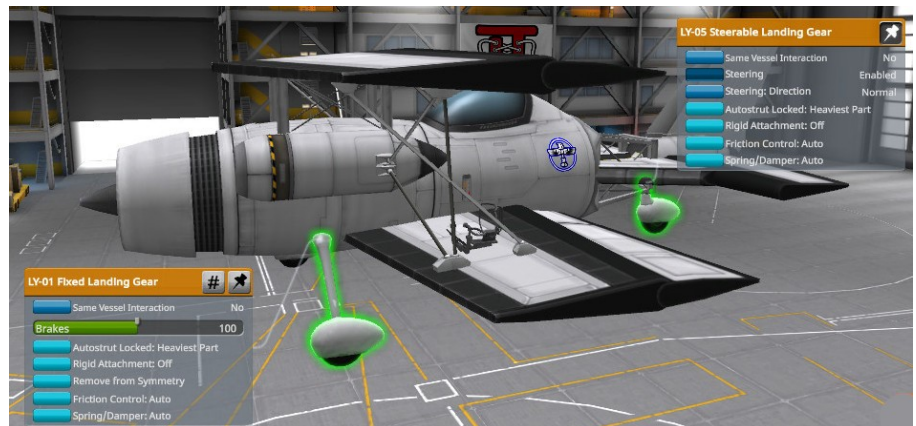
So I always fly with SAS on, but during my flight testing, I am ALWAYS flipping it off during various phases of flight to see how the airframe flies without the stabilizing assistance. I use that feedback to engineer out as much of the undesired flight characteristics as possible.

In many cases, you can't get rid of pitch/roll completely; it will come into play at some point. So the goal is to find the best middle-ground where you might see an equal amount of pitch/roll at takeoff that becomes the opposite pitch/roll at standard altitude and speed. This equalizes the job of the SAS across the full spectrum of flight and will just make your craft fly that much better. Arguments can be made for putting more emphasis on the takeoff or standard flight phase. Your preference will obviously vary, sometimes even on a craft-by-craft basis.

## Veering Off the Runway

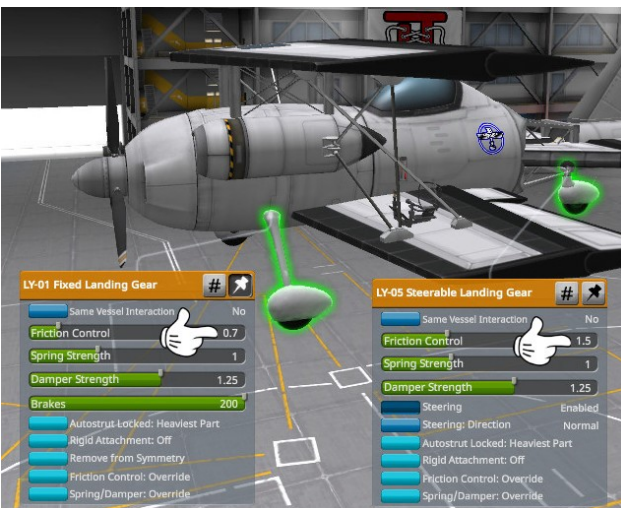
So prior to version 1.7.3, I had this silly stunt biplane design where I deliberately installed a Wheesley jet engine backwards to approximate a WAAAY overpowered propeller.

Ignoring for a moment that it could easily exceed Mach 1 in a prolonged dive, we'll just focus on the fact that it flew fine using reverse jet thrust as a propeller proxy and tracked perfectly straight down the runway for takeoff with all the default settings for landing gear as shown here.



Converting this craft to a propeller engine should be a piece of cake! ...right?

After slapping on a nifty new turboshaft engine with a cool tri-blade prop setup, I took it out on the runway. My first attempts were a complete disaster! My beloved craft that worked so beautifully with a jet engine, now veered wildly off the runway to the left every single time I tried to takeoff. My initial thought was, "Hmmm... Too much torque in the OP engine. I'm gonna have to dial that back and just live with an underpowered stunt plane." This of course, did not sit too well with me.



Adjusted landing gear friction for straighter runway tracking and better rudder yaw control during takeoff

At some point, I remembered that I solved runway tracking issues in the past by tweaking the landing gear. Not the placement of the gear, but the actual default values. The default landing gear settings have to be adjusted sometimes because as your plane gains speed on the runway, the relative airspeed increases and the nose of your plane may be getting pushed down on the runway.

This addition of downforce can put too much control input on your forward landing gear causing the craft to veer to the left or right. As noted previously, propeller craft (although slower than jets) are introducing unequal roll pressure and also have a tendency to pitch down. So there is potentially a lot more uneven pressure being applied to the forward gear during takeoff that will cause your plane to veer off the runway in one direction or another.

The solution involves adjusting the settings for friction, springs, and damper so that as the airplane gains speed, your left/right yaw is being influenced more by airflow over the rudder as opposed to the landing gear interacting with the runway. Although propeller engine designs are not as fast as jets, they introduce a lot of roll pressure disproportionately to the main (forward) gear in tail-dragger landing gear setups. By decreasing the main gear Friction Control value to 0.7 and increasing the tailwheel Friction to 1.5, we see the plane no longer veers off the runway and is easily controlled by rudder yaw input. I also tweaked damper strength a bit (set it to 1.25) to smooth out some bouncey-ness. Spring settings seemed fine at the default 1.0, but might also have to be adjusted if dealing with heavier or lighter craft.

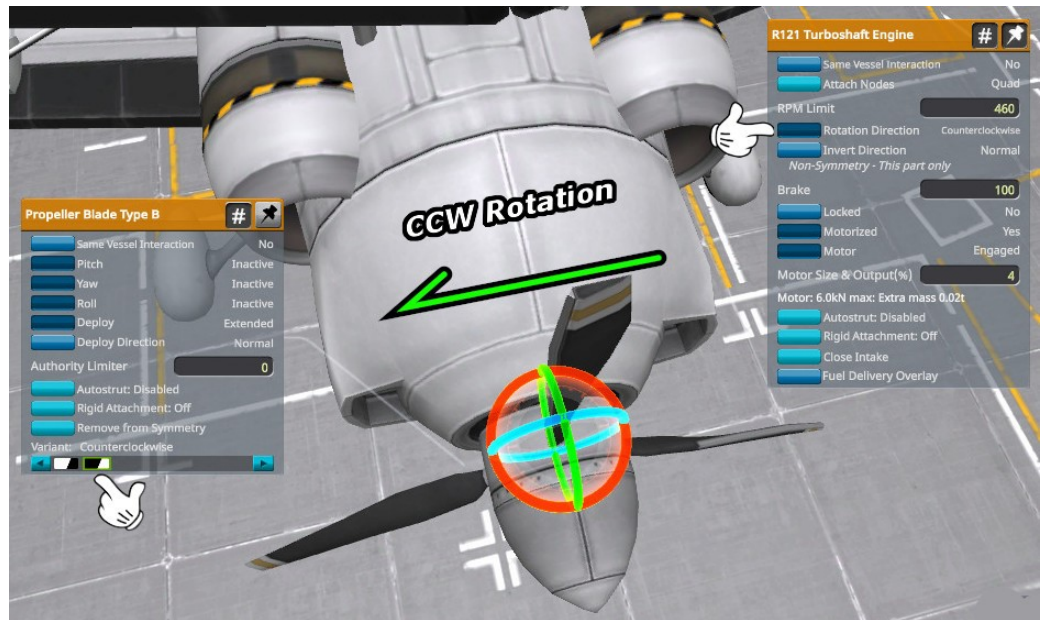
## Installing Propeller Blades

So what is the right way to install propeller blades? Below, you can see we have a turboshaft engine setup with CCW rotation. Also notice, the propeller blades have their variant type set as CCW to match. Matching them is important!

When installing the blades use the following blade settings: Pitch/Roll/Yaw should all be disabled, "Deploy" set to "Extended", and "Authority Limiter" value set to zero.

You then need to rotate the blade so it looks the way I have it here. The leading edge of the propeller blade is pointing directly into the direction of engine rotation.

Notice that the yellow/white painted tips of the blades are also correctly facing in the direction you want to go: Forward!



Correctly installed propeller blades on the real life standard CCW rotation KSP engine setup

At a complete standstill on the runway, this is the neutral or zero pitch position of the blade and regardless of how fast the rotor spins, the plane will remain stationary. For a variable pitch propeller design, this will be the default position for your airplane on the runway to start.

Assign the blade pitch (Authority Limiter) to the U/D Translate key. For this CCW engine/blade setup, increasing the blade pitch will cause the propeller to produce forward thrust, while decreasing the blade pitch will cause reverse thrust. So for takeoff, we might tap the up arrow for an Authority Value of +15 to start with wheel brakes ON. If we tap the down arrow key, we might set the value to -15 to have the plane roll backwards. I should also mention that the +/- pitch values I talk about here related to forward/reverse thrust assume you have your engine and propellers set to CCW rotation. The +/- values would be reversed if your engine and blades are setup with CW rotation.

## Managing Variable Pitch Blades

A typical takeoff might involve starting the engine and increase power to max RPMs with zero blade pitch and wheel brakes ON. Increase blade pitch to +15 and release wheel brakes to begin rolling. As speed increases, increase the blade pitch for optimal AoA producing the most forward thrust. I've noticed a pretty good kerbal rule of thumb is to just keep your blade pitch at about one-half of your stated m/s speed. So for an aircraft that rotates for liftoff at speeds in the 40-50m/s range, your blade pitch should be about 20 to 25 at rotation. You can continue to use this method during normal flight and I find that the maximum needed for kerbal world is probably about 50-ish on blade pitch with a typical max speed in the 110-ish m/s range for a single engine craft.

When coming in to land, I throttle back to about one-half or one-third power and set the blade pitch to my rotation value of 25. If I need to abort the landing, this 25 blade pitch should be a good value for climb out, and then adjust based on the "One-Half Rule" as your airspeed increases.

On short fields, once I have my wheels on the ground and committed to the landing, I adjust the prop for reverse thrust by setting the blade pitch to -15 and giving the engine full power coming to a rapid stop. This is particularly useful if you had to adjust the friction settings on your main gear and maybe don't have that wheel braking power to stop out at the Island Base runway.

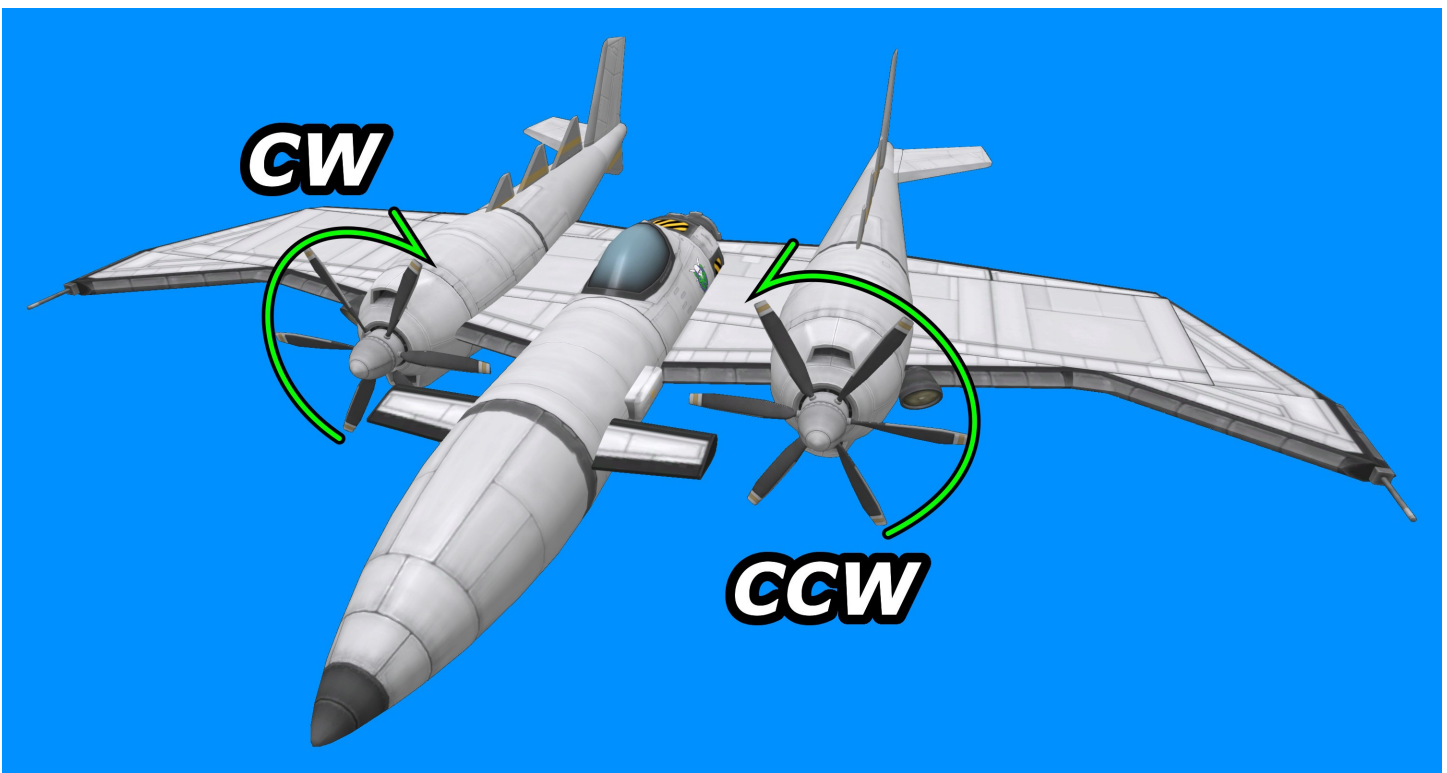
For my craft designs, I specifically setup the blade pitch values so they can only go as low as -15 for reverse and no higher than +50 for forward flight. I do this with the KAL-1000 Robotics Controller, and I'll show you how to set up your engines, blades, and rotors in the associated Engine Control Module and Variable Pitch Blade Controller sections.

## Fixed Pitch Propellers

If you just want to slap a propeller on a plane and not have to worry about maximizing the thrust for takeoff and so forth. Follow all the same rules that I mentioned previously for correct blade installation. Then just set the Authority Limiter value to something in the +25 to +40 range. You may have to flight test a bit to see which fixed pitch blade value offers the best compromise between maximum in-flight airspeed verses length of runway required to takeoff.

## Multi-Engine Designs

Setting up multi-engine designs is very much the same as our single engine discussion. A very logical approach would be to install counter-rotating engines on each wing so the roll issue we saw in our single engine craft could be easily negated. If your goal however is to create a replica craft, you might find that both engines (particularly in civil aviation) are configured to spin in the same direction in real life. "So, why is that?" I suspect it's mostly related to ease and availability of interchangeable parts. If both engines spin in the same direction, there are no left-hand or right-hand mirrored engine parts to deal with during engine maintenance and repair. In the world of kerbal however, I pretty much throw that notion out the window and say, "Go for it!"



Although there are a few notable exceptions, most counter rotating props are setup to spin toward the fuselage

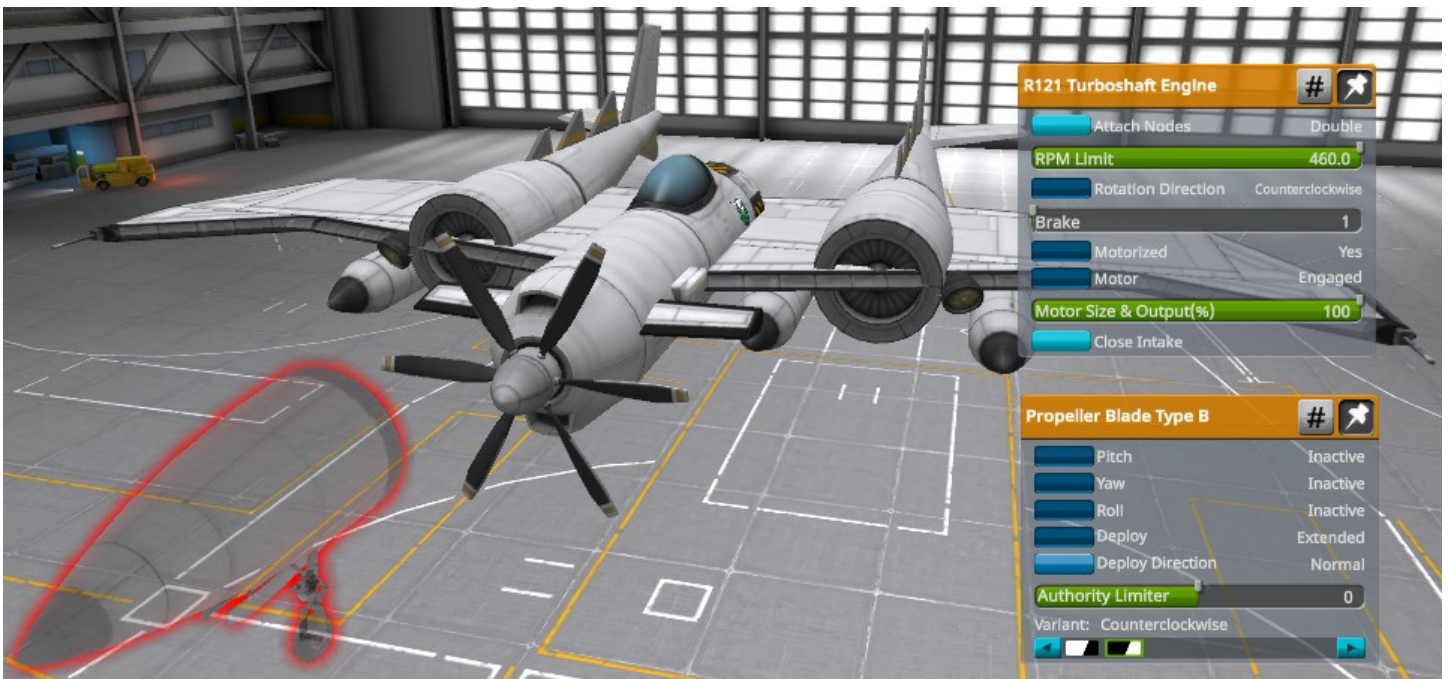
The most common rotation pattern for modern multiengine aircraft with counter-rotating engines has each engine turning toward the fuselage. So as viewed from the front, your Port (P) engine and blades would be setup with CCW rotation and your Starboard (S) engine would be CW as shown above. A notable historical example of a deviation from the above would be the P-38 Lightning, which had counter-rotating propellers that each turned away from the fuselage.

## Engine-Blade Groupings

When setting up multiengine designs like this, there is one very important design trick that you simply must know! For both single and multiengine designs, we should ALWAYS setup the motor and blades of the craft as a single unit on the longitudinal design axis using radial symmetry for blade placement.

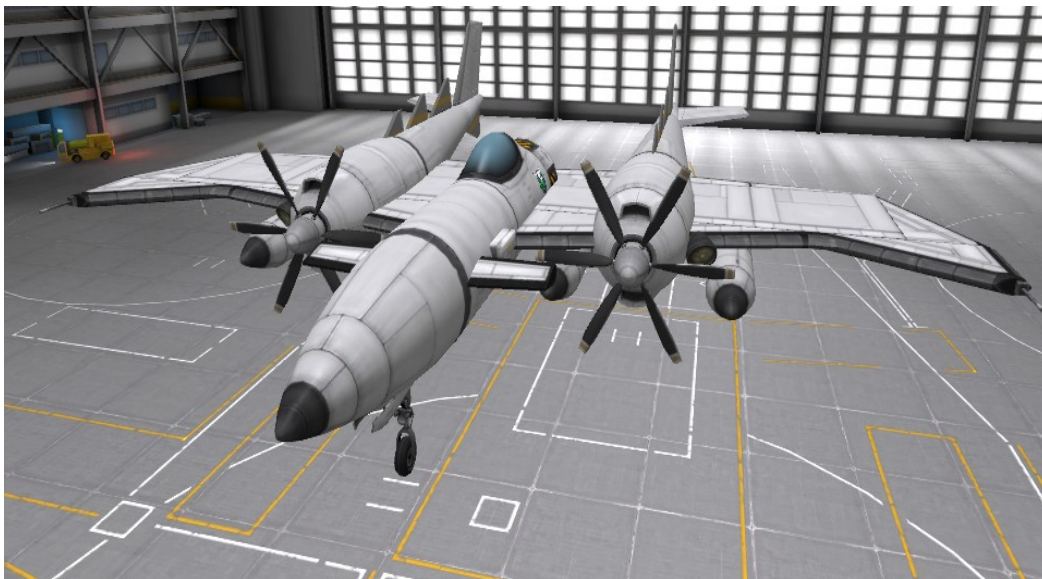
This is very important because it allows us to use radial symmetry to place our blades on the longitudinal design axis, and when we assign the blade pitch to hotkeys, we only have to reference a single blade on each engine to refer to the group! I've seen folks struggle a bit with this design element and even suggest the SPH editor needs to be modified or redesigned to do exactly what I just described. ...only because they don't know or understand the simple trick.

After setting up our CCW/CW port/starboard engines on the longitudinal design axis, we then move each engine to their respective wing nacelle positions and we replace the nose of the plane. I typically build my CCW port engine first, and copy/paste that one to the port wing by just Alt-Right Clicking on the engine. Then going back to my centered engine, I change the rotation of the engine to CW, change the blade variant to CW, and rotate one of the blades 180° to get them all facing in the right direction. The CW engine is then just moved to the starboard wing position by clicking on the engine again to pick it up and move it over.



Always setup your engine and propellers on the longitudinal design axis with radial symmetry to maintain engine-blade groupings!

Installing your engines like this is **extremely important!** Attaching the blades as a radial symmetry group lets us use a single blade for action group assignment as a representation of the entire engine-blade group. In other words, it means we don't have to place each blade individually, and we don't have to assign every single blade to a hotkey to manage blade pitch. In our sample craft shown on this page, it's the difference between having 2 key assignments (one for each engine-blade grouping), or 12 key assignments (one for each blade).



The final counter-rotating propeller setup with radial symmetry engine-blade groups on each wing nacelle

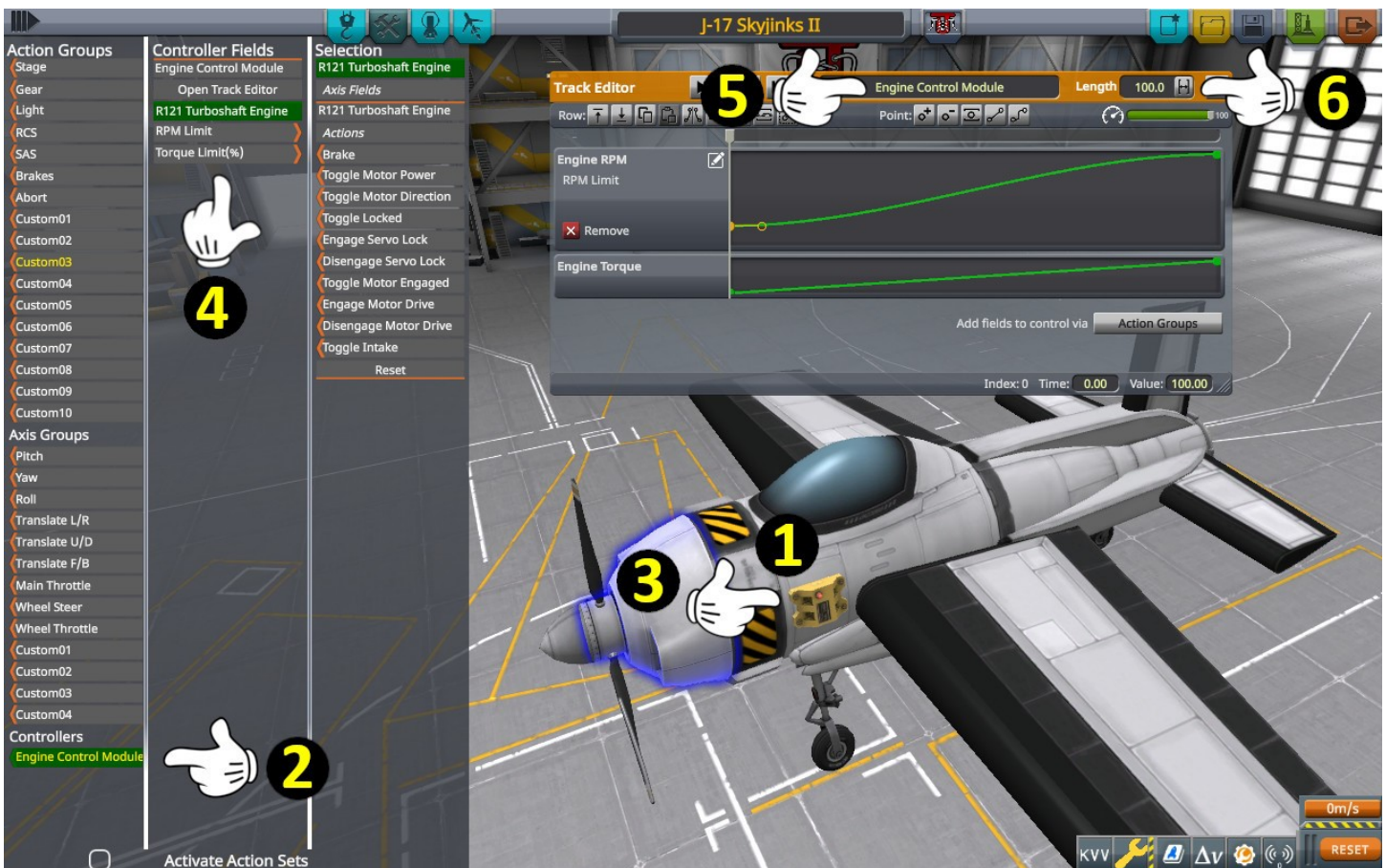
Once the engine-blade groups are in position, no more physical edits are allowed. So don't go and try to rotate a blade when the engine-group is positioned on a wing. The editor won't understand the radial symmetry if the engine is not on the longitudinal axis. If you do need to make a change, just move the engine back to the axis to make your adjustments.

## Propeller Engine Control Module (ECM)

There are two key inputs related to controlling your propeller engines that can be assigned to Action Groups: RPM and Torque. So then the question becomes, “Which one should we assign to our Main Throttle?” The answer is: both of them.

The “Breaking Ground” expansion not only introduced propellers, rotors, and engines, it also included several robotic parts and a cool new part: the KAL-1000 Robotics Controller. Engine power (torque) and RPMs are related and as we increase engine power, we should also be increasing our max RPMs. Also, when we are coming in to land and have the throttle pulled all the way back to zero or idle position, we shouldn’t see the RPMs drop all the way to zero for a dead-stick landing every single time. Interesting historical context there on the origin of the term “dead-stick” referring to the wooden propeller of WWI era biplanes coming in to land after a mission with a propeller (stick) that was no longer spinning, hence the term “dead stick landing” referring to a landing without engine power. I don’t like landing propeller craft with zero RPMs on the blades.

So we really need our throttle to control two engine inputs at the same time. We want our RPMs to range from 100 to 460 for throttle position 0 to 100. We also want our engine power (torque) to range from 0 to 100 for throttle position 0 to 100. We will use the KAL-1000 as an Engine Control Module to map both curves to the throttle for us!



1. Add a KAL-1000 controller to the fuselage wherever you like.
2. The controller will appear in the Action Groups. Select the action group item so it’s highlighted green.
3. Click on the engine (highlighted blue) so you can see the engine actions you are allowed to assign in the “Selection” column
4. Assign the engine “RPM Limit” and “Torque Limit” to the KAL-1000. You will see curves appear in the Track Editor window that you can now edit. Change the names of each curve to “Engine RPM” and “Engine Torque” respectively.
5. Change the name of the KAL-1000 to “Engine Control Module” in the Track Editor window.
6. Change the length of time from 5 to 100. You can now think of the KAL-1000 timeline as the 0 to 100 position of your throttle.

In the Track Editor window, edit the RPM curve so it starts at a value of 100 and increases with a curvilinear profile up to 460. Edit the Torque curve so it increases from 0 to 100 with a straight-line profile. Our ECM is now programmed! The final step is to just assign the “Play Position” of the KAL-1000 to our “Main Throttle” action group.



1. Select the “Main Throttle” action group so it is highlighted green.
2. Click on our ECM (highlighted blue) so you can see the actions you are allowed to assign in the “Selection” column
3. Assign the ECM “Play Position” to the Main Throttle action group.

The variable power curves for Engine RPM and Torque are now mapped to your throttle position. As you move from 0% to 100% throttle, your engine power and RPMs will ramp up based on the curves you defined in the ECM. If you throttle back to zero, your engine power goes to zero, but your propeller continues to free spin as you come in to land because RPMs were set to 100 when throttle and power are at zero. You might need to adjust the idle 100 RPM value a bit. Test fly your craft and see if 100 RPMs is sufficient to keep your propeller spinning all the way through landing. If it stops too soon, try increasing the value a bit. When you are finally ready to shutdown the engine, remember we assigned the “Engine Brake” to the 3 key with braking power set to 1 so the blade spins down slowly to a full stop.

## Variable Pitch Blade Controller

Our Engine-Group is already setup exactly as it needs to be. We can assign blade pitch to the “Translate U/D” action group and adjust the pitch as needed. However, if we just set the “Authority Limiter” value of our blades to the “Translate U/D” keys, we will be able to change the blade pitch with the full range of motion available to us. We don’t need or want the full blade pitch range from –150 to +150.

Remember, we learned that a value of –15 and +15 was where we needed the blade pitch to be for full reverse and forward power when the plane is on the runway at a standstill. We also learned from flight testing that +50 was probably all we would ever need in flight using the “One-Half Rule”. We are going to use another KAL-1000 to map the blade pitch from –15 to +50. This time, we’ll look at our multiengine example and you’ll see that our engine-group design strategy only requires that we add one blade from each engine.





1. Select the "Translate U/D" action group so it is highlighted green.
2. Click on the Variable Pitch Blade Controller (highlighted blue) so you can see the actions you are allowed to assign in the "Selection" column
3. Assign the "Play Position" to the Translate U/D action group.

Now when we are on the runway, we can control the blade pitch of both engine-blade groups by holding down or tapping on the Translate U/D keys. It's nice to have the limit values programmed into the KAL-1000 so we don't end up with useless blade pitch values outside of our desired -15 to +50 range.

I typically have two KAL-1000 controllers on my propeller aircraft so I can control the engine power and RPMs independently of the blade pitch. In the picture above, you may have noticed the ECM is mounted toward the front with the Blade Pitch Controller aft.

The ECM setup for multiengine craft is no different than my previous single engine example. You just have two engines instead of one. In the picture to the right, you can see I setup the ECM for this multiengine craft the same as my single engine example. I just have duplicate RPM and Torque curves for the P and S engines.

Programming coaxial multiengine designs would be setup the exact same way. You would just have two of your engine-blade groups on the same axis in your SPH design. The KAL-1000 programming would be exactly the same you just might refer to the Front and Rear engines instead of Port and Starboard.



## Installing Rotor Blades

Installing rotor blades is really no different than installing propeller blades. All the same settings apply when installing rotor blades: Pitch/Roll/Yaw should all be disabled, “Deploy” set to “Extended”, and “Authority Limiter” value of zero. You then need to rotate the blade so the leading edge is facing in the right direction for your engine rotation.

Standard rotor blade rotation is not as clear-cut as the standard CCW setting for prop engines. When viewed from overhead, American helicopters almost always spin in a CCW direction.

On the other hand, many European rotors spin in a CW direction. I tend to prefer setting up propellers and rotors with CCW rotation because I like the orientation of the +/- blade pitch values.

It is interesting that the view perspective changes to “view from above” when talking about helicopter rotors, and now the KSP references for CW and CCW are not backwards. So I guess Squad was just doomed to have the default rotation backwards for either airplanes or helicopters. LOL!



Properly installed rotor blades and motor settings for a CCW engine/rotor setup

## Collective and Cyclic Helicopter Controls

I have seen quite a bit of confusion related to stock rotor blade setup. How do you make rotor cyclic controls work? ...and a lot of, “Why doesn’t my helicopter fly?” discussion. I must say, it is very curious that the default settings for rotor blades are not the same as propeller blades because for all practical purposes, they behave exactly the same way and should be treated as such.

The control options for rotors in the current (1.7.3) implementation do **NOT** allow for true helicopter-like controls. Most notably, the cyclic control is non-existent. Although, you might be fooled into thinking a cyclic has been implemented because the default settings for rotor blades have Roll, Pitch, and Yaw enabled! These settings really should be viewed in the context of slapping the rotor blade on the side of an aircraft fuselage and treating it like any other wing or aileron-type control surface. The design idea, in that respect, I do not find to be particularly useful. Not to mention, the cyclic control of a helicopter really has nothing (from a pilot control standpoint at least) to do with yaw.

The cyclic control of a helo is reading the position of the yoke stick and is tilting all the blades length-wise, in a cyclical fashion, as they rotate around the shaft. The mechanical part in the rotor assembly that is responsible for this is the swashplate. The cyclic control relates to the forward/backward and side-to-side movement. It’s not related to yaw, which is typically controlled by a tail rotor. We just do not have a simple stock cyclic control implementation in KSP.

What we do have is the ability to control rotor blade pitch (just like propeller blades) which is essentially the collective control or up/down lifting force. This is managed by setting the “deployed” state of the rotor blade and controlling the pitch angle of all rotor blades collectively. Again, we do this by adjusting the Authority Limiter value the same exact way we controlled the variable pitch propellers. You can assign this to a key and once again the translate style keys behave properly for this purpose. Alternatively, if you read-on, you might find my preferred approach of mapping it all with the KAL-1000 works even better!

## Kerbo-Kopters!

In kerbal world, we can overcome the missing cyclic control in a wonderfully kerbal-esque fashion by adding reaction wheels (control gyros) to use as both a pseudo-cyclic controller and yaw stabilizer. We can certainly add a functional tail rotor for yaw, but mapping the exact Torque/RPM/Pitch required to precisely counter the same on your main rotor is going to be challenging at best. Even if you carefully mapped your tail rotor power curve with the KAL-1000, we are still going to need the control wheel gyro for the lack of cyclic. So the tail rotors are at least semi-functional, if not entirely accurate for perfect yaw control.



Now that doesn't mean that our rotary flying contraptions are mislabeled helicopters, but they don't really fully approximate their real life counterparts. I've taken to calling these things Kerbo-Kopters and the main difference between them and real life helos is the pseudo-cyclic control is tilting the ENTIRE craft for both forward/reverse pitch and side-to-side movement; whereas a real life Helo would be applying tilt with the cyclic to the spinning rotor plane via the swashplate.

*"Could a cyclic or swashplate type feature be added?"* Yeah, I s'pose so, but it's probably a level of mechanical complexity that will further confound the folks who are already confounded by the perceived difficulty in setting up the individual blades that we have to play with now. Hinges that act like a swashplate on the rotors would seem to be a very difficult programming task. Possibly including some sort of omni-directional tilting ability on the motor shaft where the blades are attached might be doable. Even so, you'd have a challenging time trying to perfectly manage your tail rotor yaw control without some sort of control wheel gyro assistance.

So why make it any more difficult? Given the level of helicopter-ish control that I was able to achieve in my designs so far, I don't really see a need to make it any harder for folks. I'm quite happy with them exactly as they are now. I can build helos that pretty much act and fly very similar to real helos without the need for cyclic control complexity that might make a stable hover unusually difficult to achieve.

## Coaxial Rotor Designs

Coaxial helicopters use a pair of counter-rotating blades on the same axis to negate the effects of unwanted yaw that would otherwise be present. The designs are most notable in real life for the complexity of the coaxial rotor shaft and the absence of tail rotors. The nice thing about creating them in KSP is the complexity is pretty darn low and we still get to ditch the tail rotor!

One mistake I made early-on was to connect the top motor directly to the spinning shaft of the bottom motor. My initial thought was, *"No problem, I'll just make the motor on top spin twice as fast!"* Although the net effect was a top rotor that made 230 net RPMs, the fact that it had to spin at 460 RPM while the bottom was turning at 230 caused an imbalance where the top rotor was having a significant impact on yaw that the bottom rotor at 230 RPM could not match. So I had to separate the two motors and attach each to the fuselage separately. I used an octagonal strut to attach the top motor to the base of the non-spinning base of the lower motor. This allowed me to set the RPMs equivalent for the top and bottom engines. The yaw effect was equalized and the craft now flies perfectly without a tail rotor.



Make sure each motor in a coaxial design is independently attached to the fuselage or a non-spinning part of the lower motor. The octagonal strut in this design was attached to the non-spinning part of the lower engine.

## Helicopter Engine Control Module (ECM)

After reading through the previous sections on how I setup propeller ECM and Blade Pitch with separate controllers. It would seem logical that the same format would work with a helicopter. We could control engine RPM and Torque with the throttle and now the blade pitch is just equivalent to the collective control of a helicopter. ...and that would be very true. Perhaps setting up something like that using the Translate U/D keys as your collective would be your preference.

After a considerable amount of flight testing, I found that with helicopters, I was able to blend the collective pitch into the throttle position along with RPM and torque. ...and what the heck, might as well throw some jet engine thrust in there while we're at it! The result is a single ECM that can efficiently manage both lift and forward thrust across the full helicopter flight spectrum. We only need to use one KAL-1000.

The process for programming the ECM is no different than what we learned previously with motors and blades. Now we are just going to blend all those curves together in a single controller, and assign the "Play Position" to the throttle. You remember, or can look back to review how we added power curves to the ECM. I'm not going to restate all that again, but I will show you every power curve I added to the ECM and talk you through the logic that was applied to the various power curves.



The helicopter ECM simultaneously manages power curves for 7 different flight systems. We control them all with our Main Throttle!

In the above screen capture, I numbered the rows 1 through 7 for each curve and will explain how each curve was setup below. The length of time for the Track Editor window was once again set to 100 so we can think of it like 0-100% throttle. I also added droplines for the significant throttle positions: 0%, 50%, 70%, 80%, and 100% to further aide my explanations.

Flight testing revealed the rotor blades like to hover at +15 pitch and will ascend rapidly at +50 pitch. Initially, a zero or neutral blade pitch was tried during descent. The relative speed of the wind coming up through the blades continued to produce lift which resulted in very slow descending characteristics. To counter the slow descent more aggressive negative blade pitch angles were tried and a -20 pitch was deemed optimal for descending. It is also worth noting again that pitch values are inverted when referencing CW engine-blade groups. Notice items 1 and 2 below are identical pitch patterns but the CW values are inverted.

1. Upper Blade Pitch: 0% -20, 70% +50, 80% +50, 100% +15 (normal values CCW)
2. Lower Blade Pitch: 0% +20, 70% -50, 80% -50, 100% -15 (inverse values CW)
3. Upper Engine Torque: 0% 0, 50% 100
4. Lower Engine Torque: 0% 0, 50% 100
5. Upper Engine RPM: 0% 200, 70% 460, 80% 460, 100% 230
6. Lower Engine RPM: 0% 200, 70% 460, 80% 460, 100% 230
7. Juno Jet Engines: 0% 0, 80% 0, 100% 100

Throttle range 0-80% for normal vertical flight ascend/descend without jet thrust, with maximum vertical lift achieved in the 70-80% band. Throttle range 80-100% rapidly spools up the Juno Jet engines for high-speed horizontal flight. As jet thrust is added, blade pitch is reduced to +15 for hover and the rotors are now configured to act like wings during forward flight.

Max vertical lift is achieved at 80% and max speed at 100%. During high speed flight the rotor RPMs are reduced by 50% to 230 RPM. This reduces the stabilizing gyro effect of the rotors and allows for much better roll control at speed. Helicopter engine torque curves are more aggressive on the low end of the throttle compared to propeller engine profiles. This gives the helicopter pilot a lot more low-end torque to make very responsive hover adjustments when coming in to land.

## Update Addendum

Since original publication, there have been several updates to KSP which have changed some of the underlying properties and adjustments for props, rotors, and motors. I'll try to list here any deviations that I notice over time.

### KSP 1.8 Update

I noticed the 1.8 update introduced a few changes to propellers and rotors but nothing too drastic:

- The 1.8 upgrade introduced new mapping for rotor blade pitch
- The function of Authority Limiter was changed and Deploy Angle was added for pitch control
- In 1.7.3 the range of motion was -150 to +150 and the new range is -22.5 to +22.5 or exactly 15% of the prior values
- The ECM (KAL-1000) just needs to be reprogrammed to use Deploy Angle instead of Authority Limiter and changing the values to be 15% of the previous values

### KSP 1.9 Update

The rotors now have cyclic control. In the process they also inverted the thrust pitch vectors for clockwise rotation blades and props. So in the section of the manual where I talk about inverting the KAL-1000 mapping for CW and CCW rotation; that is no longer required. The blade pitch mapping works the same now for both.

I should warn however... the cyclic control is terrible. I don't recommend using it and would stick to the kerbo-kopter design I described in this guide where you are basically just using the rotor collective to adjust lift. I'd stick to using reaction wheels for yaw and pseudo-cyclic. You can create something that sorta works with the cyclic, but you will never get to a stable hover like you see in my helo designs. All comes down to whether or not you want something that is actually useful for missions in KSP or just a toy to try to land on a helipad with considerable effort mixed with frustration.