



# رباتیک هوایی

استاد : دکتر امین طلائی زاده

پروژه رباتیک هوایی

تاریخ: 1403/03/18

اعضای گروه:

عرفان رادفر

شماره دانشجویی: 99109603

محمدجواد شمس الدین سعید

شماره دانشجویی: 99106266

اسما حلاجی

شماره دانشجویی: 99109585

بهار 1403

2	صورت پروژه
3	انتخاب پرنده
4	روابط دینامیکی
6	شبیه سازی در Simulink
7	مدل Simscape
9	هاور در شرایط اولیه
10	افزایش سرعت 10 درصدی تمام روتورها
11	افزایش سرعت 10 درصدی نصف روتورها
12	حرکت با زاویه pitch برابر با 30 درجه
14	تغییر جهت نصف روتورها
15	کنترل (امتیازی)
15	کنترل زوایای اویلر و ارتفاع
18	کنترلر مکان

## صورت پروژه

### تعریف مسئله

در فاز اول پروژه هدف شبیه سازی مولتی روتور می باشد. به منظور ایجاد چالش و خلاقیت، از شما خواسته می شود تا یک مولتی روتوری را طراحی کنید که کوادکوپتر ساده نباشد. بدین منظور مراحل زیر را داریم:

#### شبیه سازی دینامیکی

- پرنده مورد نظر را انتخاب کنید
- روابط دینامیکی حاکم بر حرکت مولتی روتور را استخراج کنید.
- سپس این روابط را در نرم افزار متلب/سیمولینک شبیه سازی کنید.
- حال به کمک کتابخانه **simcape** مدل طراحی شده پرنده را وارد محیط سیمولینک کرده و به رفتار پرنده را صورت سه بعدی شبیه سازی کنید.
- سرعت دورانی روتورها را به گونه ای تعیین کنید که پرنده در شرایط اولیه خود ثابت بماند.
- سرعت دورانی همه روتورها را نسبت به حالت هاور 10 درصد کاهش دهید و رفتار پرنده را بررسی کنید.
- سرعت دورانی نصف روتورها (به انتخاب شما بر حسب پرنده) 10 درصد افزایش دهید.
- حال یک تابع برای روتورهای پرنده طراحی کنید که پرنده حدود 30 درجه زاویه پیچ بزند و در آن زاویه بماند. (در این حالت پرنده باید به سمت جلو حرکت کند).
- حال در صورتی که جهت دوران نصف روتورهای پرنده (به انتخاب شما بر حسب پرنده) برعکس شود، بررسی کنید اگر پرنده بخواهد به سمت جلو و به سمت جلو چپ حرکت کند، چه رفتاری خواهد داشت.

#### کنترل (امتیازی)

- برای پرنده یک سیستم کنترل زاویه و ارتفاع از نوع کنترلر PID مستقل برای رول، پیچ، یاو و ارتفاع طراحی کنید. بدین منظور کافی است مقدار لحظه ای زوایای رول، پیچ و یاو و همچنین ارتفاع را فیدبک گرفته و از بلوک PID سیمولینک استفاده کنید. سپس زوایای 0، 30 و 45 درجه را برای رول، پیچ و یاو به عنوان زاویه ی دلخواه بدهید و رفتار پرنده را بررسی کنید.
- سپس یک کنترلر مکان برای سیستم طراحی کنید. برای این کار، کافی است از ساختار کنترلی **Cascade** استفاده کنید. در این سیستم کنترلی، خروجی کنترلر مکان، زاویه دلخواه رول و پیچ است.

#### شرایط

- وزن کلی پرنده را 5 کیلوگرم در نظر بگیرید. با فرض اینکه واحد مرکزی پرنده 0.5 کیلوگرم باشد، با تقسیم کردن وزن باقی مانده به تعداد روتورهای انتخابی و بدست آوردن وزن هر روتور، ماتریس ممان اینرسی را به صورت تحلیلی بدست بیاورید و استفاده کنید.
- ضریب تراست  $C_T$  را 0.02 و ضریب گشتاور  $C_M$  را 0.005 در نظر بگیرید.
- دستگاه اینرسی و بدنی را به صورت NED در نظر بگیرید.
- حالت اولیه پرنده هاور در ارتفاع 10 متر از سطح زمین است.
- برای حل روابط سینماتیک از کواترنیون استفاده کنید و برای نمایش نهایی دوران از زوایای اوپلر با ترتیب ۳۲۱ استفاده کنید.
- هیچ گونه شرایط اغتشاش در پرنده و محیط وجود ندارد.

با توجه به آنکه می‌خواهیم مولتی روتوری طراحی کنیم که بتواند نسبت به یک کوادکوپتر معمولی توانایی عملکرد بیشتری داشته باشد، کوادکوپتری با قابلیت tilt روتورها در قسمت جلوی پرنده را انتخاب می‌کنیم. توانایی این پرنده در این است که می‌تواند برای حرکت به جلو، بدون نیاز به آنکه بدنه آن pitch بزند، روتورهای خود را به جلو خم کند و به جلو شتاب بگیرد. در عین حال برای yaw میتواند دو موتور جلویی خود را در خلاف جهت خم کند که سبب می‌شود سریعتر مانور yaw را انجام دهد و مانورپذیری پرنده بالاتر رود. هر چند این مسئله سبب نیاز به استفاده از دو موتور servo در دو روتور جلویی می‌شود، اما برای استفاده‌های خاص صنعتی و بخصوص در پرنده‌ای با ابعاد بزرگ، این نوع دینامیک می‌تواند مقرون به صرفه باشد و جلوی مانورهای بالای بدنه را بگیرد.

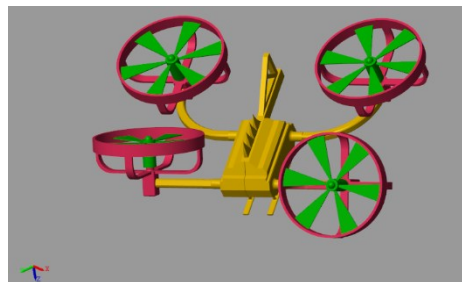
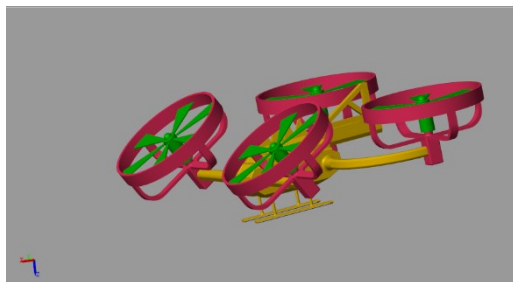


Figure 1: مانور yaw و حرکت به جلو

همچنین این نوع ساختار را در پرنده‌های عمودپرواز که قابلیت تبدیل به حالت بال ثابت را دارند نیز می‌بینیم. اما در این پرنده‌ها، به علت موتور احتراق داخلی روتورها، کنترل کاملی که در مولتی‌روتورها وجود دارند را نداریم. همچنین می‌توان از این ساختار برای تاکسی‌های پرنده که بین آسمان خراش‌ها و موانع انسانی پرواز می‌کنند و در عین حال بخاطر آسایش سرنشینان نمی‌توانند pitch داشته باشند، استفاده کرد.



Figure 2: پرنده های tilt rotor و ایده تاکسی هوایی درون شهری

برای روابط دینامیکی سیستم، از روابط کوادکوپتر ساده، با استفاده از quaternion و ساختار 321 بهره می‌گیریم؛ با این تفاوت که باید برای روتورهای جلو، دوران‌های  $\gamma_1$  و  $\gamma_2$  را نیز در نظر بگیریم. در نهایت به روابط زیر می‌رسیم.

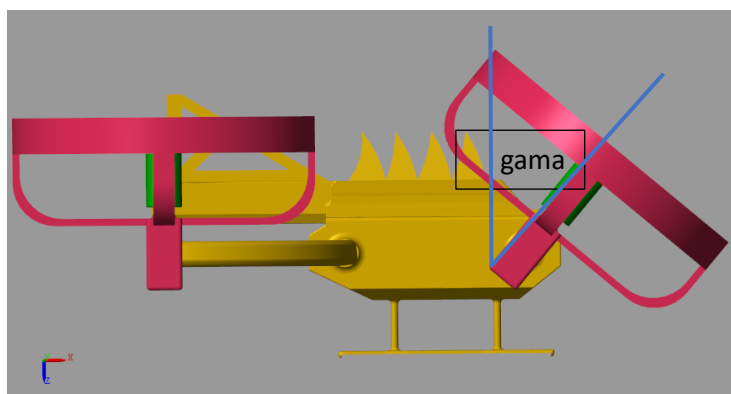


Figure 3: زاویه گاما

$${}^b_F = \begin{bmatrix} T_1 \sin \gamma_1 + T_2 \sin \gamma_2 \\ 0 \\ -(T_1 \cos \gamma_1 + T_2 \cos \gamma_2 + T_3 + T_4) \end{bmatrix}$$

$${}^b_M = \begin{bmatrix} b(-T_1 \cos \gamma_1 + T_2 \cos \gamma_2 + T_3 - T_4) + \frac{c_m}{c_T} (-T_1 \sin \gamma_1 + T_2 \sin \gamma_2) \\ b(-T_1 \cos \gamma_1 + T_2 \cos \gamma_2 - T_3 - T_4) \\ \frac{c_m}{c_T} (-T_1 \cos \gamma_1 + T_2 \cos \gamma_2 + T_3 - T_4) + b(-T_1 \sin \gamma_1 + T_2 \sin \gamma_2) \end{bmatrix}$$

این معادلات، معادلات مربوط به نیرو و گشتاور در دستگاه بدنه می‌باشند. همچنین این متغیرها را بر حسب نیاز برای mixer تعریف می‌کنیم.

$${}^b_M = A \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \quad A = \begin{bmatrix} -bC_1 - \frac{c_m}{c_T} S_1 & bC_2 + \frac{c_m}{c_T} S_2 & b & -b \\ bC_1 & bC_2 & -b & -b \\ -bS_1 + \frac{c_m}{c_T} C_1 & bS_2 - \frac{c_m}{c_T} C_2 & \frac{c_m}{c_T} & -\frac{c_m}{c_T} \end{bmatrix}$$

$${}^b_F = B \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \quad B = \begin{bmatrix} S_1 & S_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -C_1 & -C_2 & -1 & -1 \end{bmatrix}$$

همچنین برای به دست آوردن ممان اینرسی پرنده، با فرض اینکه بدنه اصلی 0.5 کیلوگرم و روتورها در مجموع 4.5 کیلوگرم یعنی هر کدام 1.125 کیلوگرم باشند، ممان اینرسی تقریبی پرنده به صورت زیر خواهد بود. (در اینجا فرض شده است که tilt روتورهای جلویی تقریباً تاثیر زیادی در I سیستم ندارند و تقریباً ثابت هستند.)

چون سیستم در راستای x تقارن دارد،  $I_{xy}$  و  $I_{zy}$  صفر است. همچنین چون طرف جلو و عقب پرنده دقیقاً ساختار یکسانی ندارند،  $I_{xz}$  نیز مقدار مثبتی دارد. تقریباً برابر با 0.2. همچنین برای  $I_{xx}$ ،  $I_{yy}$ ،  $I_{zz}$  مقادیر مثبت، 1 و 1.5 را در نظر میگیریم. دلیل اینکه چنین مرتبه ای برای اینرسی های قطری در نظر گرفتیم آن است که هر روتور به تقریب اینرسی دورانی برابر با  $0.5^2 = 0.28$  دارند که 0.3 تا 0.5 متر همان طول عمود بازوهاست. پس در مجموع  $0.28 \times 4$  برای هر کدام از اینرسی های قطری در نظر میگیریم. اینرسی  $I_{zz}$  برای آن بیشتر است که باید فاصله بازو از مرکز جرم به جای فاصله عمود استفاده شود. در نهایت به ماتریس دورانی زیر می‌رسیم که در شبیه سازی استفاده شده است.

$$I = \begin{bmatrix} 1 & 0 & 0.2; \\ 0 & 1 & 0; \\ 0.2 & 0 & 1.5 \end{bmatrix};$$



برای شبیه سازی دینامیکی، معادلات ذکر شده در قسمت قبل را وارد تابع Plant در فایل model\_0.slx می کنیم و حلقه های بازگشتی آن را به فرمت زیر متصل می کنیم. به این شکل سیستمی خواهیم داشت که روابط دینامیکی سیستم را اجرا کند.

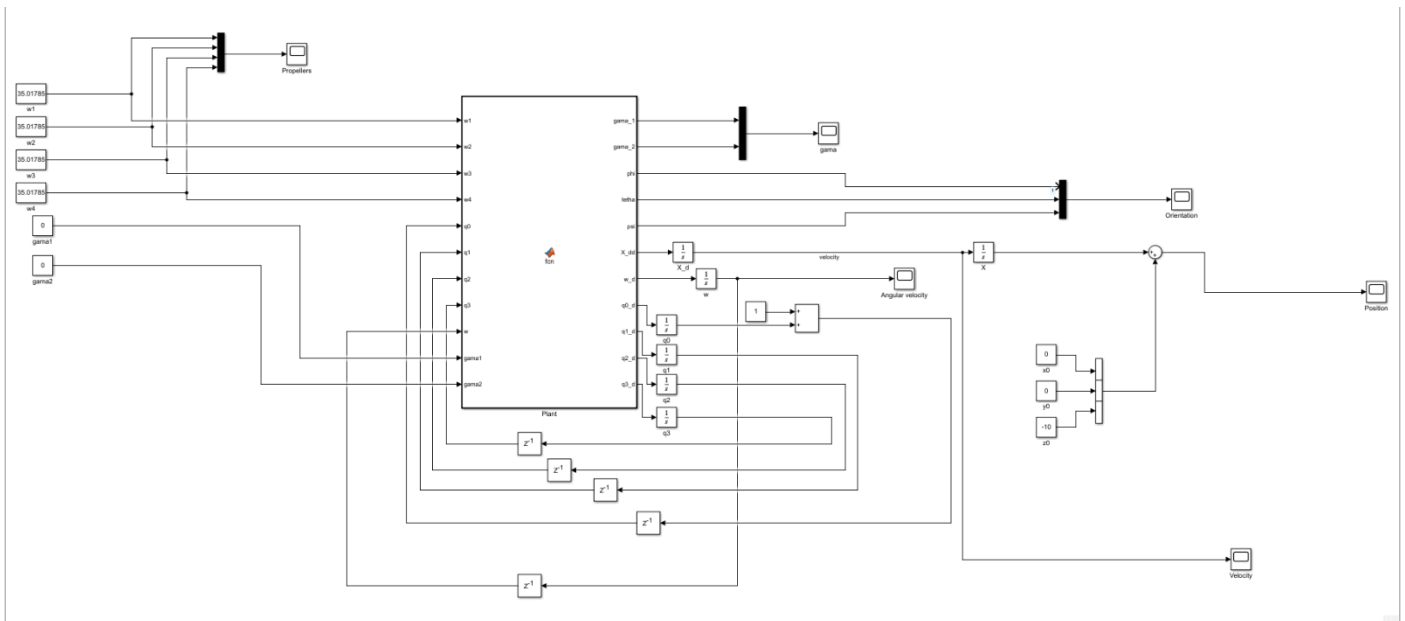


Figure 4: مدل دینامیکی ساده سیستم

در این شکل 6 ورودی داریم که شامل دور موتورها و زاویه tilt دو موتور جلویی است. همچنین برای شماره گذاری روتورها و جهت چرخش از ساختار زیر استفاده می کنیم.

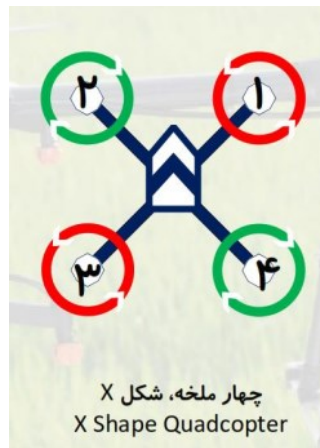


Figure 5: شماره گذاری روتورها

برای طراحی مدل *simscape* از طراحی زیر استفاده می‌کنیم. برای آنکه امکان *tilt* روتورهای جلویی در راستای  $y$  داشته باشیم، روتورهای 1 و 2 را به صورت عمودی به بدنه متصل کرده‌ایم. باید خروجی‌های دینامیک سیستم را به مدل *simscape* بدهیم که این *subsystem* به شکل زیر است.

در این مدل زاویه‌های *tilt*، سرعت روتورها، موقعیت و جهت پرنده بر اساس زاویه‌های اوپلر 123 به مدل داده می‌شود چرا که مفصل *bushing* به ترتیب حول  $x$ ،  $y$  و  $z$  دوران می‌کند.

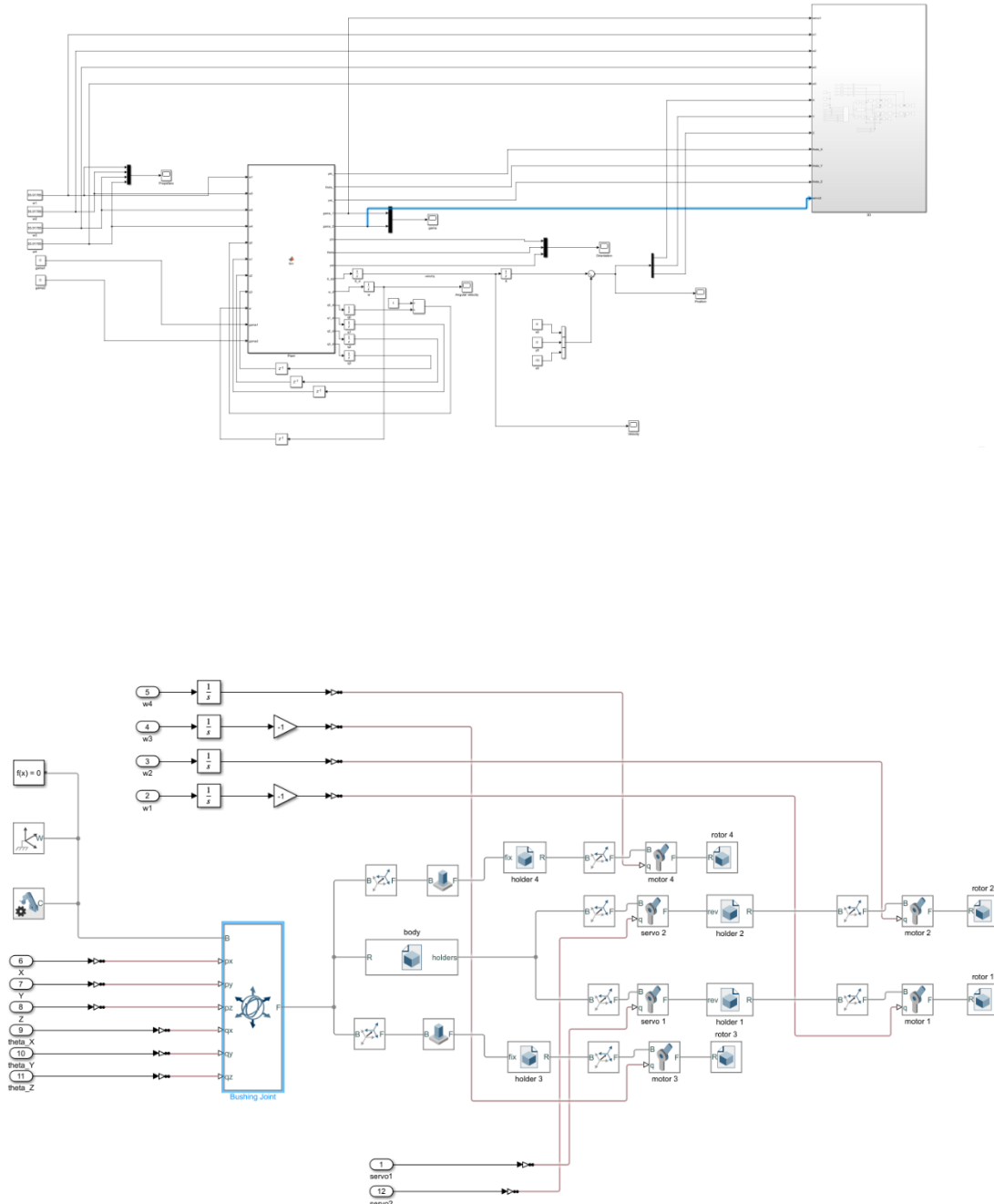


Figure 6: subsystem برای اضافه کردن مدل *simscape*



همچنین در طراحی مدل پرنده از نقشه شماتیک زیر برای ابعاد استفاده شده است.

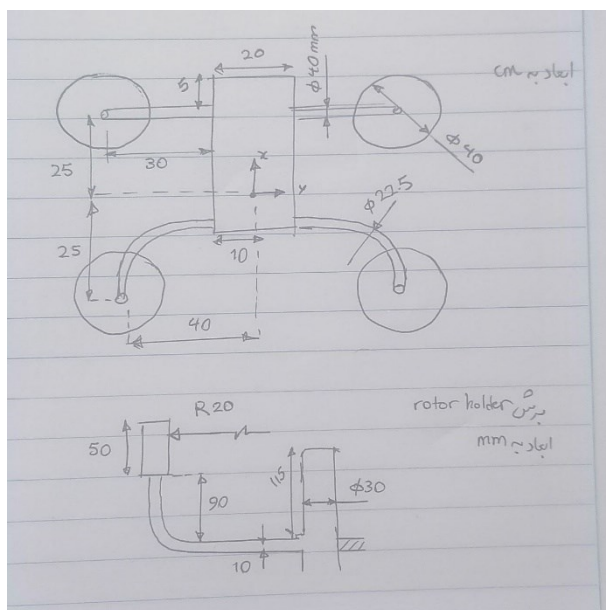


Figure 7: نقشه پرنده

در نهایت تصویر کلی مدل به صورت زیر خواهد بود.

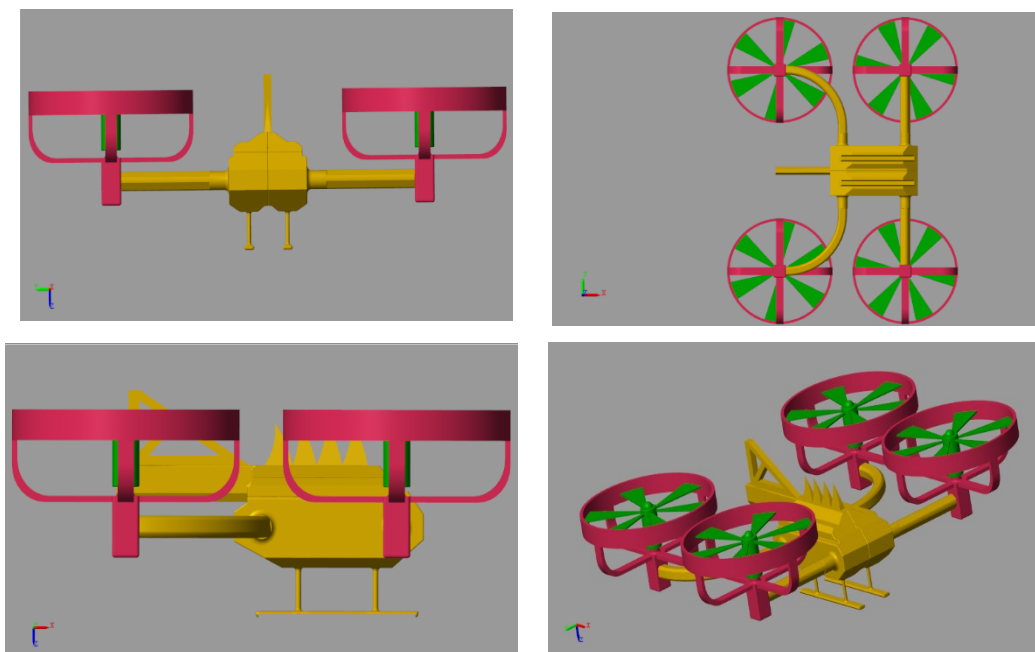


Figure 8: شمای کلی پرنده

برای آنکه در ارتفاع 10 متری یعنی  $z=-10$  ثابت بمانیم، باید زاویه tilt را صفر کنیم و همچنین، باید مجموع تراست تمام روتورها، برابر با وزن کل پرنده شود. با توجه به مشخصات خواسته شده برای پرنده، سرعت دوران را 35.01785 رادیان بر ثانیه به دست می‌آید. در نمودار position محورهای 1، 2 و 3 به ترتیب محورهای  $x, y, z$  هستند. همانطور که انتظار می‌رود پرنده ثابت می‌ماند، همچنین جهت آن تغییر نمی‌کند. (model\_0\_1)

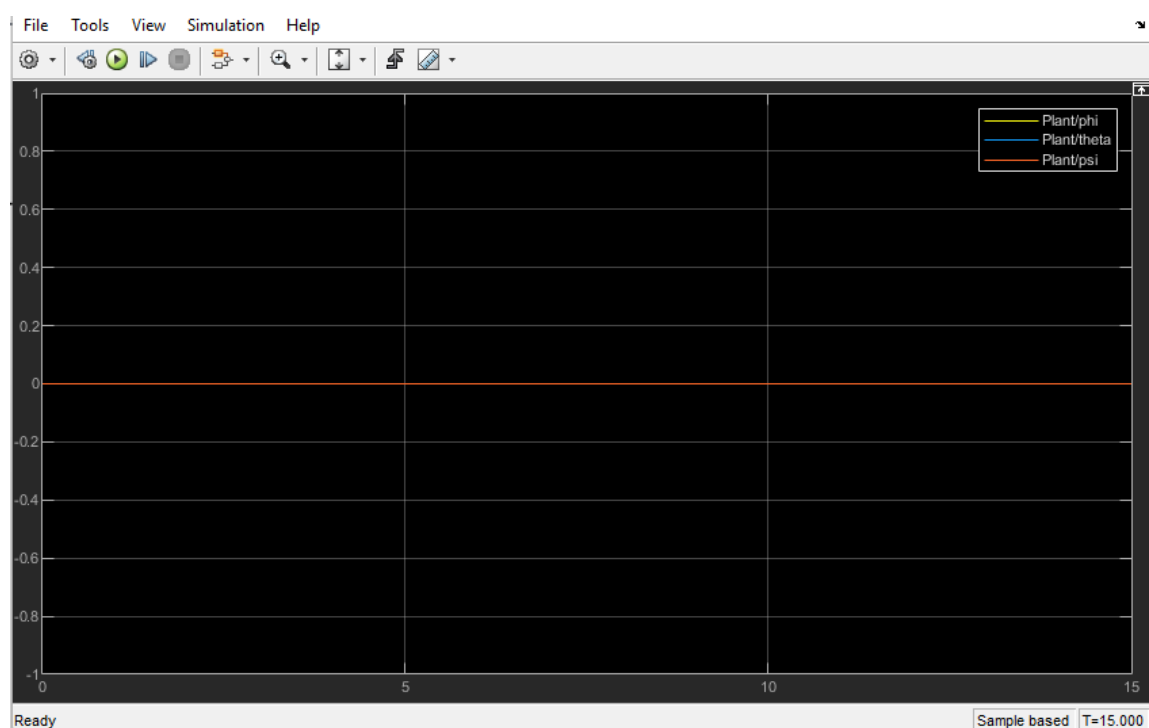
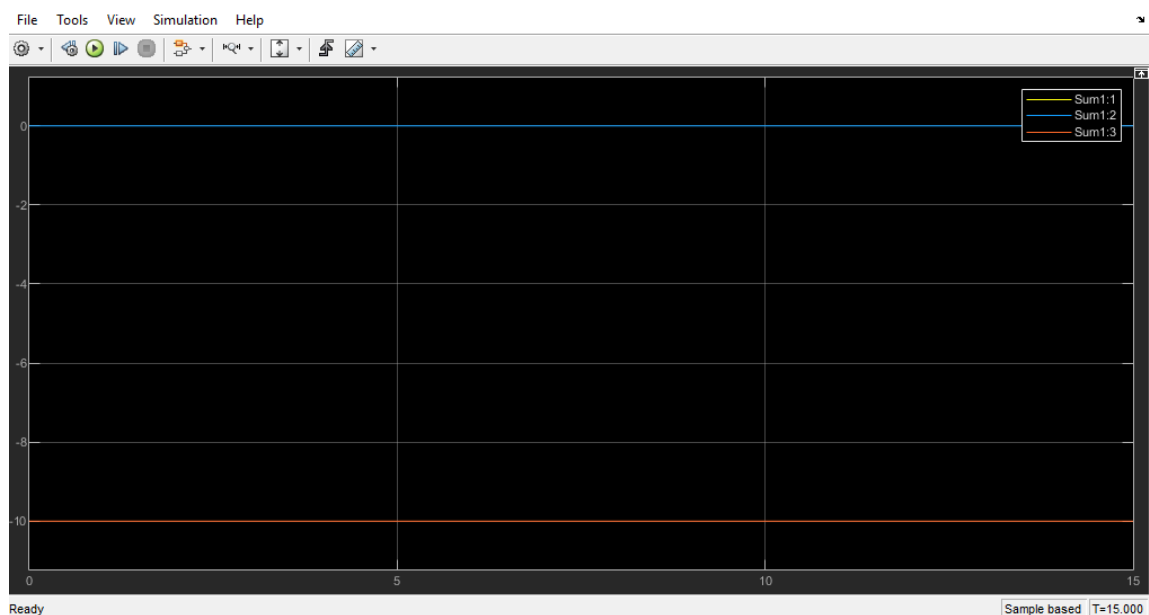


Figure 9: شبیه سازی هاور در شرایط اولیه

## افزایش سرعت 10 درصدی تمام روتورها

در این حالت پرنده باید به علت شتاب خالص، ارتفاع بگیرد (Z آن کاهش یابد) در عین حال نباید تغییر جهتی ایجاد شود. در نمودارهای زیر همین پدیده مشاهده می‌شود. (model\_0\_2)

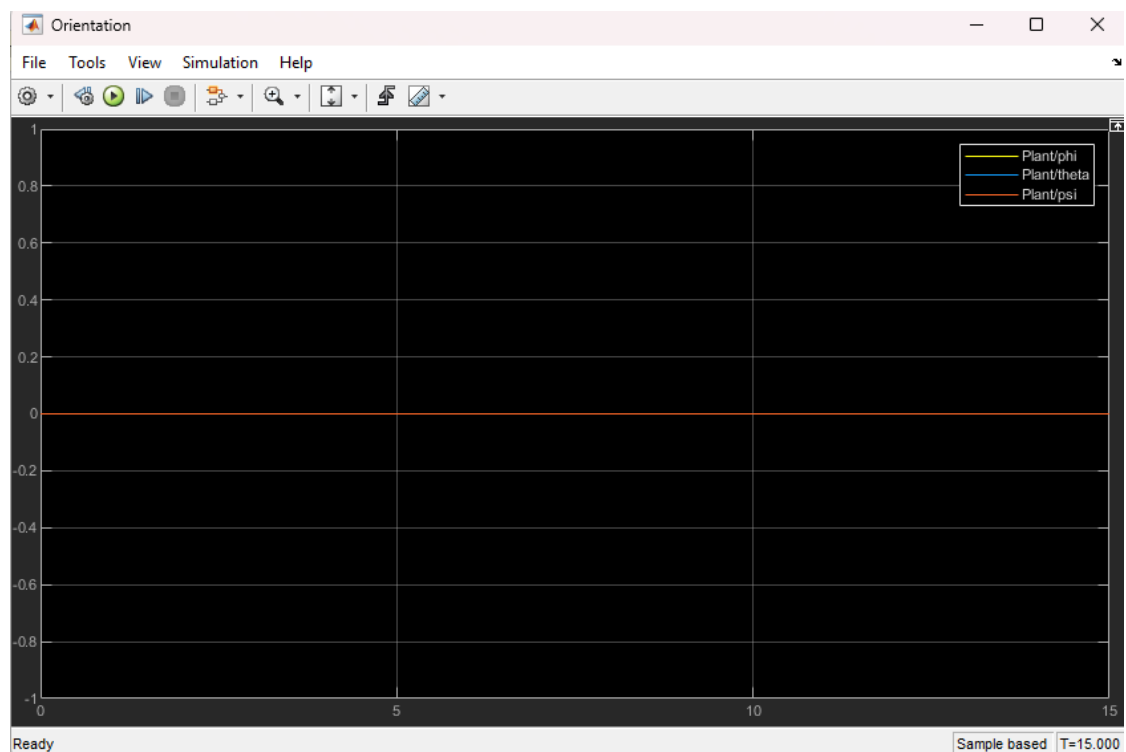
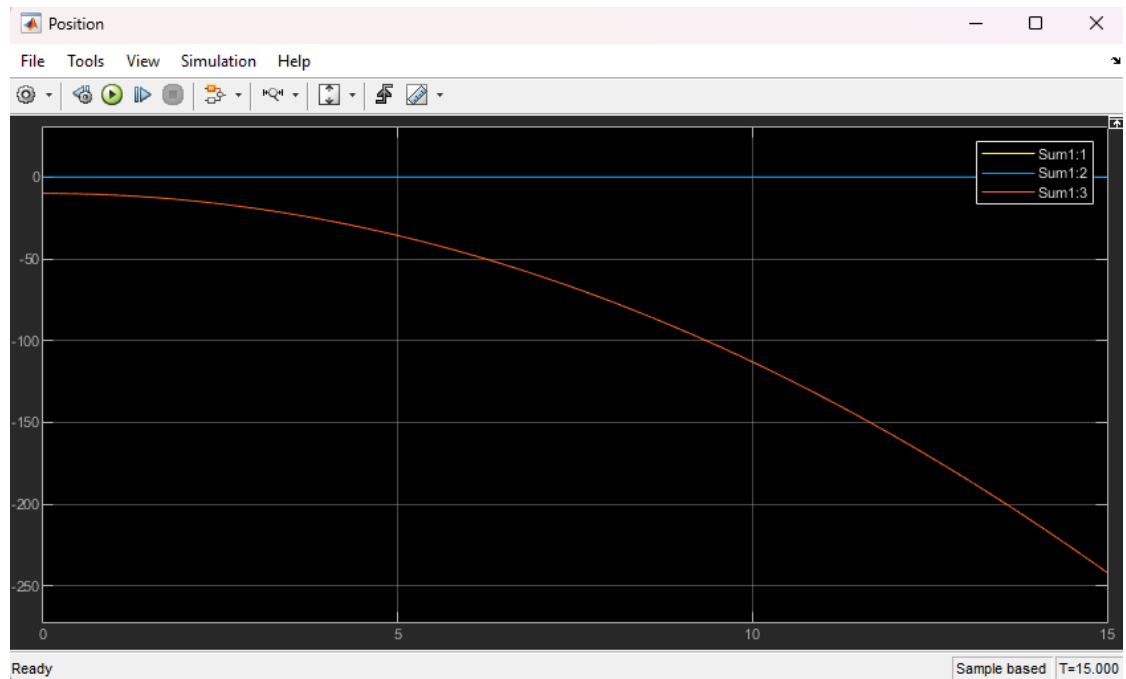


Figure 10: افزایش 10 درصدی سرعت تمام روتورها

## افزایش سرعت 10 درصدی نصف روتورها

در اینجا دو روتور 1 و 2 را انتخاب می‌کنیم. در ابتدا شاهد شتابگیری زاویه‌ای به علت ایجاد pitch خواهیم بود. عملاً تنها تغییر زاویه انجام شده نیز در راستای  $\gamma$  است. اما به علت همین پیچ پرنده به جلو حرکت خواهد کرد و در گذر زمان سرعت چرخیدن آن حول محور خودش بیشتر می‌شود. اما هیچ حرکتی در راستای  $\gamma$  نخواهیم داشت. همچنین بخاطر چرخیدن نیروی خالص زیادی به پرنده در راستای Z وارد نمی‌شود که همین سبب سقوط آن خواهد شد. این رفتار در نمودارهای زیر قابل مشاهده است. (model\_0\_3)

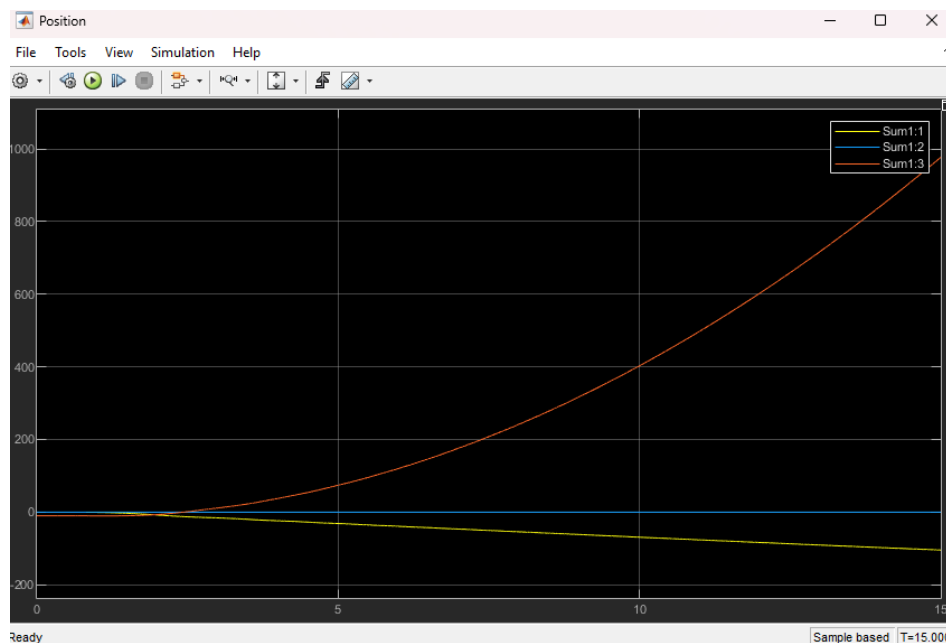
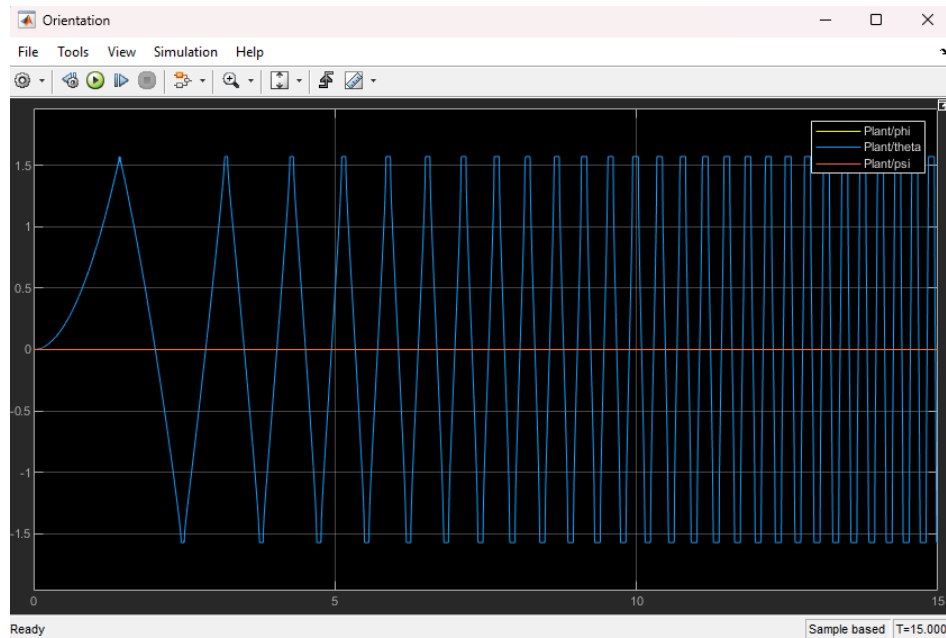


Figure 11: افزایش 10 درصدی سرعت نصف روتورها

## حرکت با زاویه pitch برابر با 30 درجه

برای حرکت در زاویه ای خاص باید بتوان آن زاویه را فارغ از شرایط موجود ( اغتشاش های سیستم و باد) حفظ کرد. در حالت ایده آل و با فرض اینکه بخواهیم کنترلر مدار باز طراحی کنیم، باید گشتاوری ابتدا مثبت و سپس با همان طول زمان، منفی در راستای  $\gamma$  اعمال بکنیم، تا پرنده در زاویه 30 درجه ثابت شود. اما این روش به علت عدم وجود **feedback** بسیار ناپایدار خواهد بود، به همین دلیل یک کنترلر PID برای این هدف طراحی می کنیم که در بخش های بعدی به آن اشاره خواهد شد. اما فعلا با روش مدار باز و با استفاده از **signal builder** خواسته مورد نظر را برآورده می کنیم، به این شکل که سرعت روتورهای جلویی را به اندازه 0.3 و به مدت 2 ثانیه زیاد می کنیم و سپس به مدت همان 2 ثانیه، سرعت آنها را تقریبا به همان میزان کم می کنیم. چون تغییرات سرعت روتورها کم است، گشتاور ایجاد شده در هر دو حالت تقریبا به صورت خطی افزایش و کاهش می یابد و سرعت دورانی نهایی مولتیروتور صفر خواهد شد. (model\_0\_4)

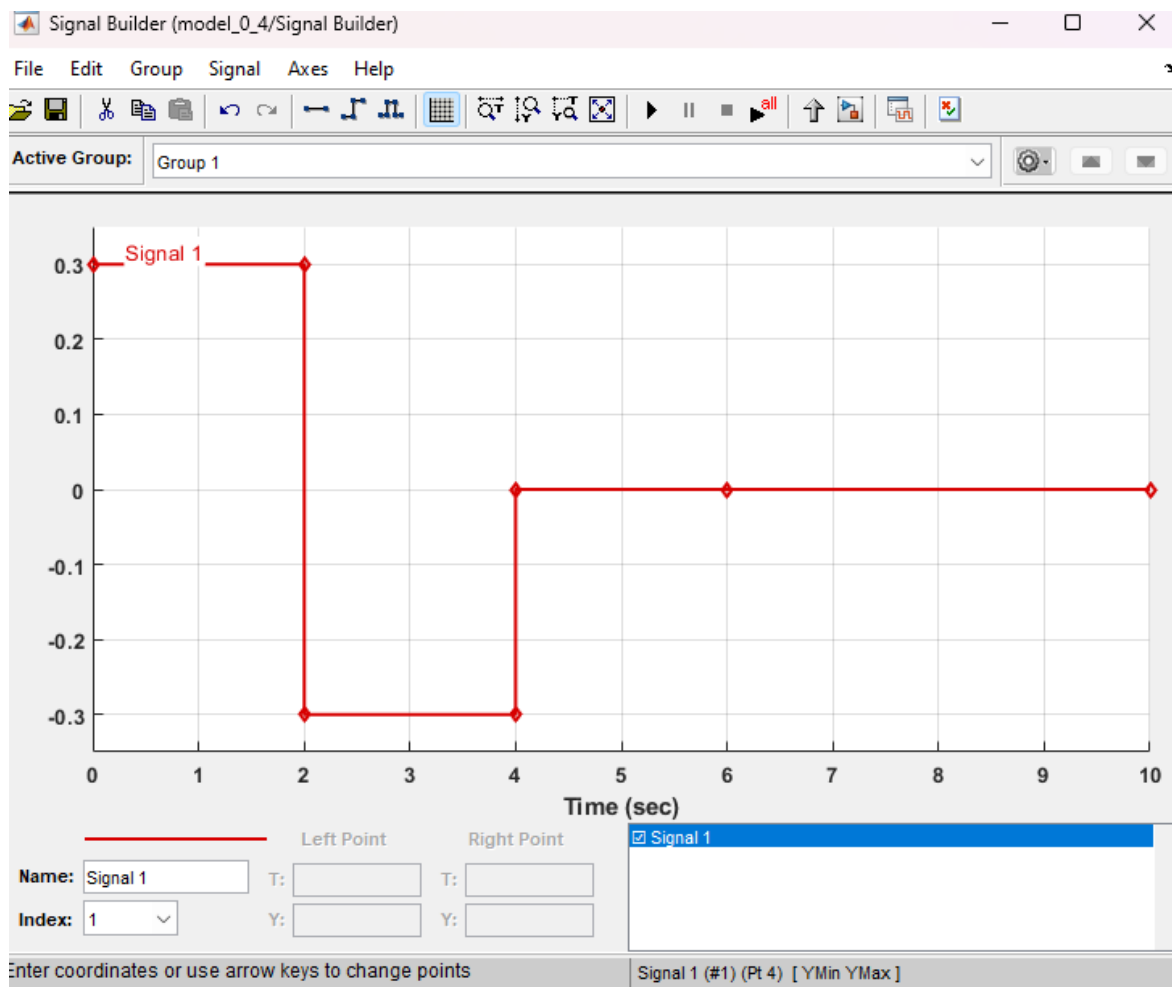
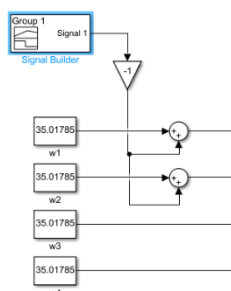


Figure 12: تغییر سرعت روتورها با signal builder



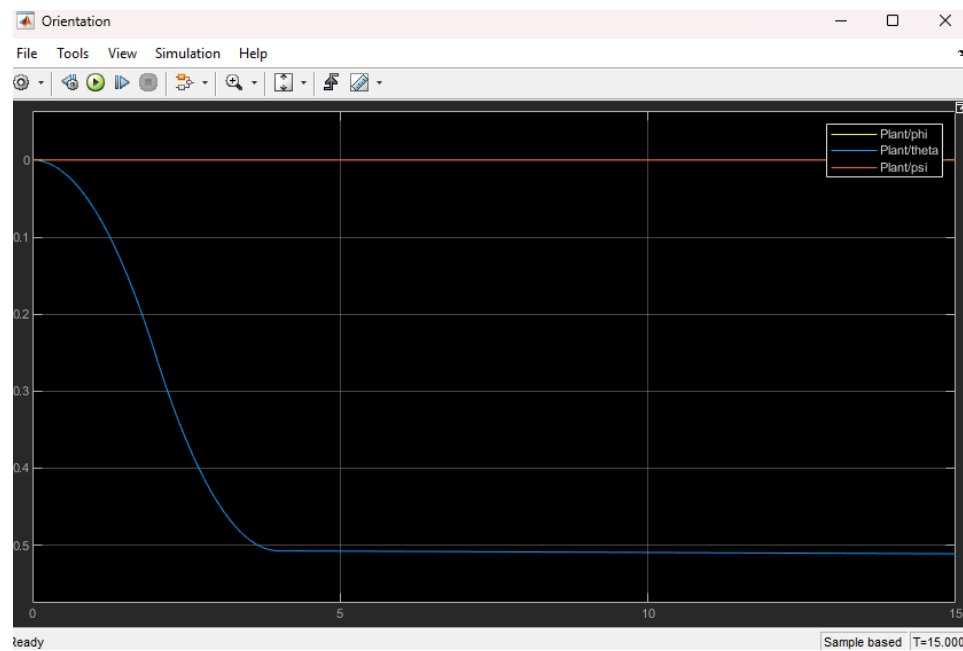
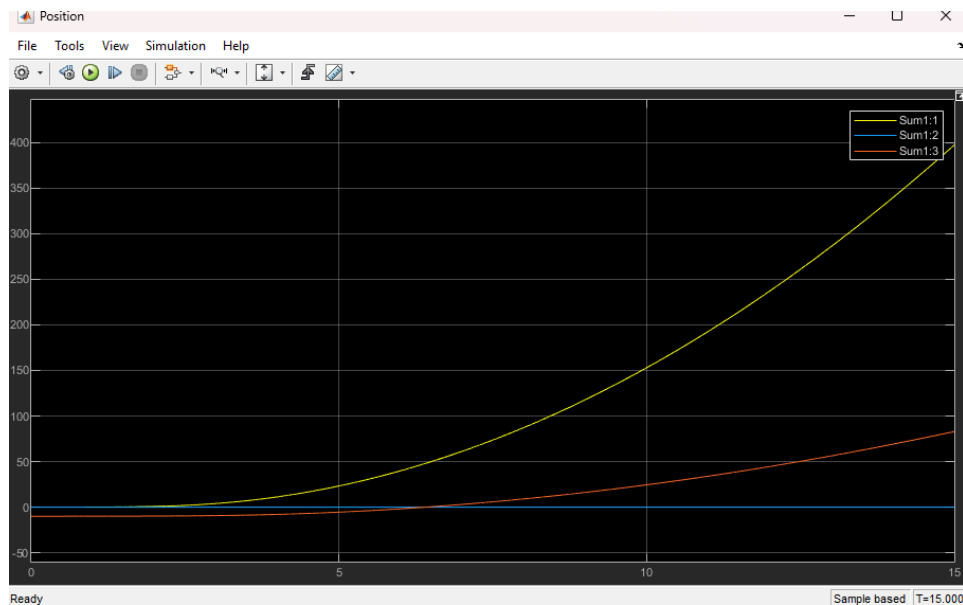


Figure 13: دوران  $pitch$  با زاویه  $30$  درجه

همانطور که مشاهده می‌شود، پرنده به اندازه  $30$  درجه ( $0.52$  رادیان) به سمت جلو (در جهت منفی  $y$ ) خم می‌شود و در همان نقطه می‌ماند. به این علت به سمت جلو (مثبت  $x$ ) حرکت می‌کند. در عین حال چون سرعت روتورها به حالت اولیه باز می‌گردد و حالا تمام نیروی آنها به سمت بالا نیست، در نتیجه پرنده به سمت زمین شتاب می‌گیرد. می‌توان با افزایش دور تمام روتورها به میزان یکسان، دوباره به سمت بالا شتاب گرفت و پرنده تنها در راستای  $x$  حرکت می‌کند.

در اینجا فرض می‌کنیم که دو روتور جلویی تغییر جهت داده اند ولی به این صورت که گشتاور ناشی از گردش موتورهای برعکس شده ولی تراست همچنان به سمت بالاست. برای اینکار تمام جملات معادلات دینامیکی  $M$  این دو روتور که دارای ضریب  $C_m$  هستند را قرینه می‌کنیم. در این حالت برای هاور تغییری پیش نمی‌آید. همچنین زمانی که می‌خواهیم به جلو حرکت کنیم نیز مشکلی نداریم چرا که گشتاور دو روتور جلو که باید کم شوند، هنوز خلاف یکدیگرند و خنثی می‌شوند. اما در حرکت به سمت طرفین، دچار مشکل می‌شود. در این حالت باید روتورهای سمت راست و چپ با سرعت‌های متفاوتی کار کنند. در حالتی که جهت چرخش آنها نیز مخالف است؛ پس  $yaw$  ایجاد می‌شود و پرنده نمی‌تواند حرکت مستقل برای جابه‌جایی همراه با رول داشته باشد. در اینجا این پدیده نمایش داده شده است. (model\_0\_5)

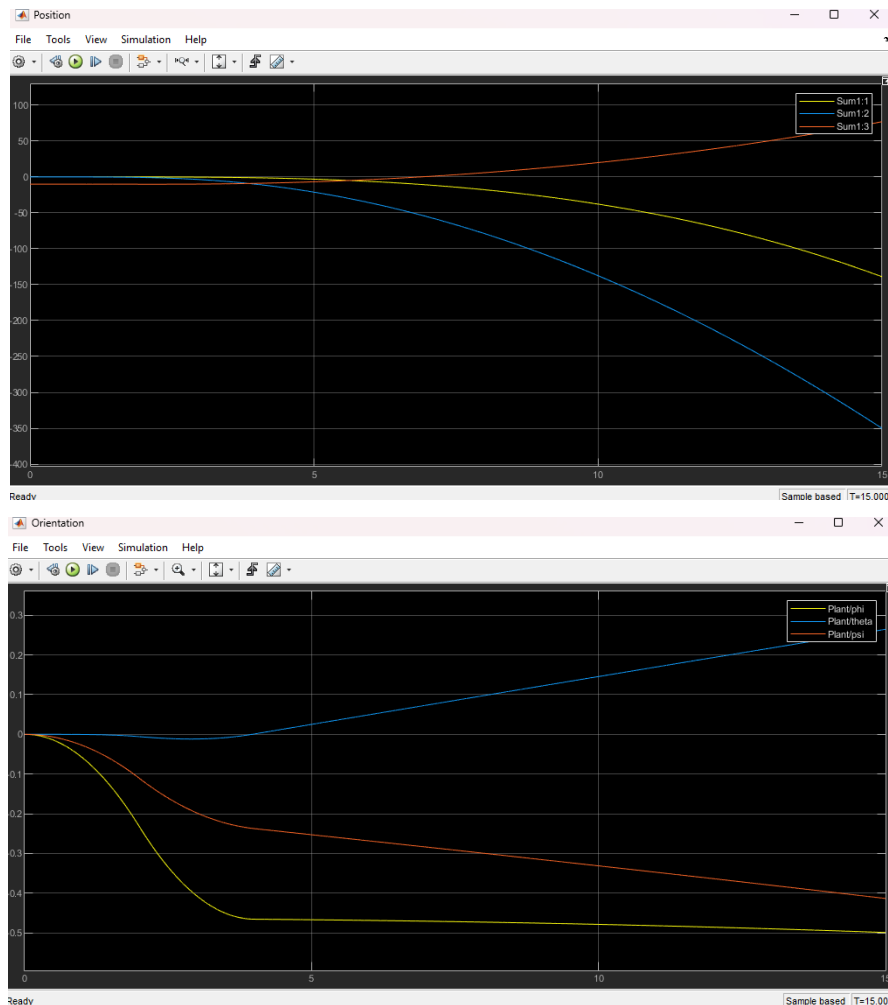


Figure 14: تغییر جهت نصف روتورها

همانطور که مشاهده می‌شود، با اینکه تنها دو روتور 1 و 4 تغییر زاویه داده اند و زاویه گاما نیز صفر است، تمام زوایای اوایلر تغییر می‌کند. هرچند اگر از کنترلر برای تعیین زاویه گاما استفاده کنیم، می‌توان اثر این پدیده را کاهش داد.



## کنترل زوایای اوایلر و ارتفاع

برای کنترل خواسته شده ابتدا باید یک mixer طراحی کنیم. یعنی باید بتوانیم برای  $M$  دلخواه، ورودی روتورها و زاویه‌های tilt را تعیین کنیم. چون معادلات دینامیکی سیستم داری  $\sin$  و  $\cos$  گاما می‌باشد، باید به روش عددی این معادلات را در زمان حل کنیم. برای این منظور از روش تکرار استفاده می‌کنیم. به شیوه الگوریتم زیر:

$$d_1^0 = \varepsilon \quad d_2^0 = \varepsilon \quad \text{while } d_2^n > 10^{-3} \ \& \ d_1^n > 10^{-3}$$

$$\begin{aligned} d_1^n &= \gamma_1^{n+1} - \gamma_1^n \\ d_2^n &= \gamma_2^{n+1} - \gamma_2^n \\ T &= \begin{bmatrix} A \\ B \end{bmatrix}^{-1} \begin{bmatrix} M \\ F \end{bmatrix} \end{aligned}$$

$$\gamma_1 = \text{atan2}(^b f_x - T_2 S_2, -^b f_z - T_3 - T_4 - T_2 C_2)$$

$$\gamma_2 = \text{atan2}(^b f_x - T_1 S_1, -^b f_z - T_3 - T_4 - T_1 C_1)$$

در نهایت با انجام تکرار به مقدار صحیح برای گامای 1 و 2 می‌رسیم. البته حد 20 را برای هر بار محاسبه گاما و  $T$  قرار می‌دهیم تا شبیه‌سازی قطع نشود. سپس سراغ طراحی ساختار کنترلی می‌رویم. (model\_1)

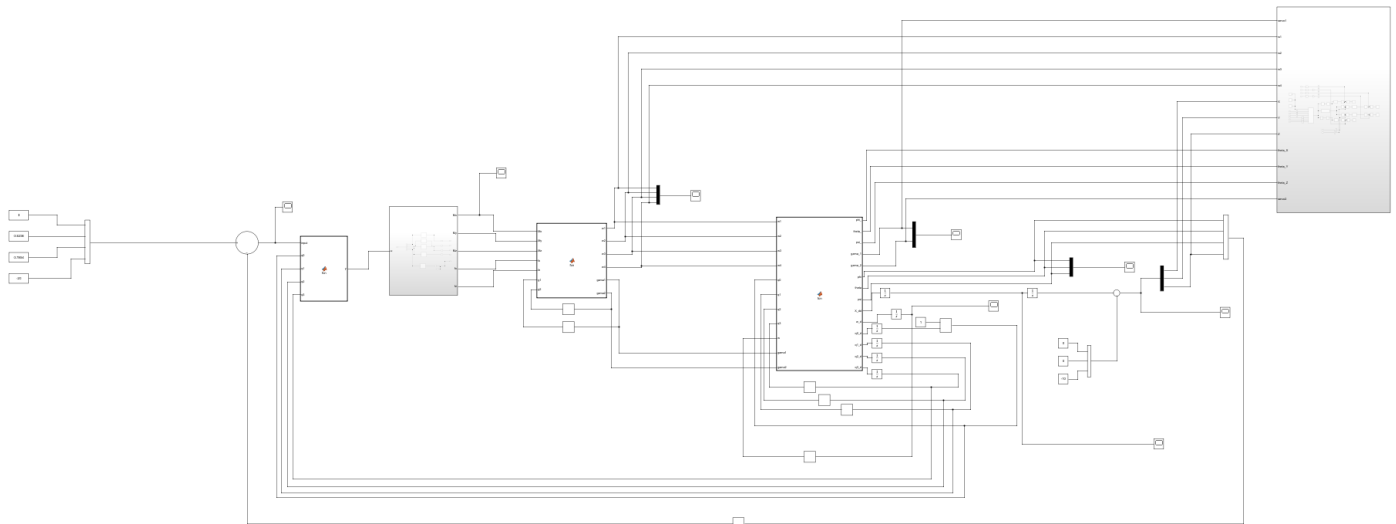


Figure 15: سیستم و کنترلر ارتفاع و زاویه

در اینجا ارتفاع سیستم و زوایای اوایلر به عنوان فیدبک گرفته می‌شوند و خطای آنها ابتدا به دستگاه بدنه منتقل می‌شوند. سپس این خطا به کنترلر داده می‌شود تا نیروهای اعمالی و گشتاور مورد نیاز را به mixer بدهد. ساختار کنترلر بر اساس PID بوده و با روابط زیر کار می‌کند. چون مولتی روتور هیچوقت نمیتواند نیروی در راستای  $y$  بدنه وارد کند، معادله بدیهی  $f_y=0$  را حذف می‌کنیم و به 5 معادله می‌رسیم. در اینجا می‌توان به جای سرعت‌ها، خود جابه‌جایی را به عنوان خطا به کنترلر داد.

$$\begin{aligned} f_z &= -mg + PID(\hat{v}_z - v_z) \\ f_x &= PID(\hat{v}_x - v_x) \\ M_x &= PID(\hat{v}_y - v_y) \\ M_y &= PID(\hat{\theta} - \theta) \\ M_z &= PID(\hat{\psi} - \psi) \end{aligned}$$

در این معادلات، خطا به بلوک PID مستقل داده شده و دستورهایی مورد نظر بعد از عبور از فیلتر اشباع، به میکسر ارسال می‌شود. همچنین بخاطر غیرخطی بودن سیستم مولتی روتور، می‌توان به صورت شهودی و با سعی و خطا ضرایب PID به دست آورد.

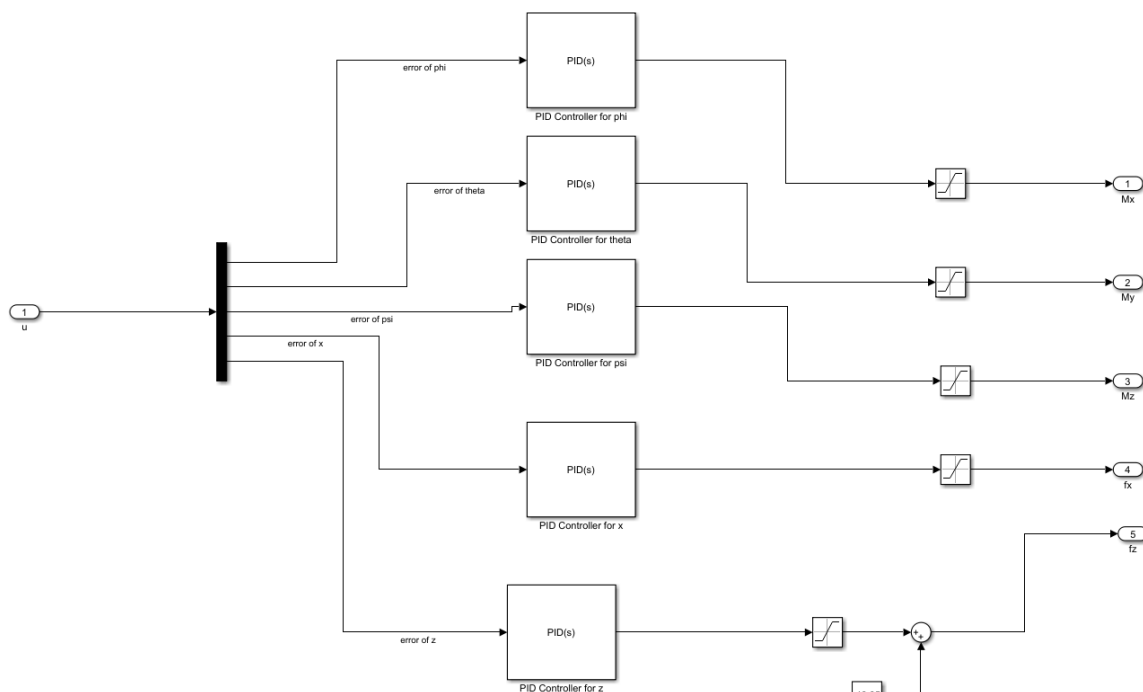
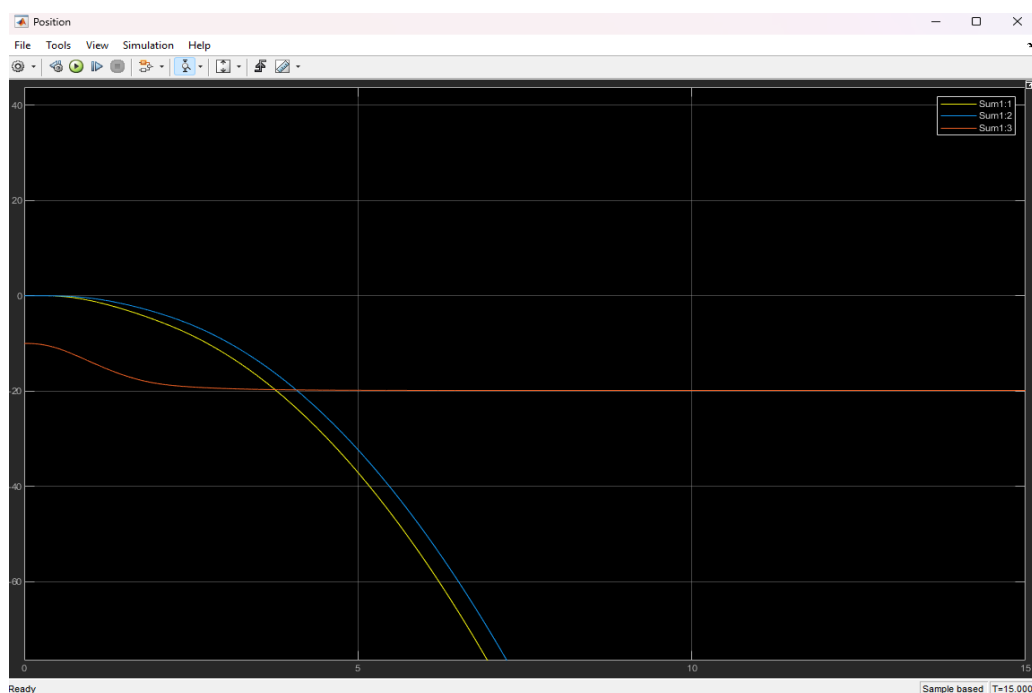


Figure 16: کنترلر PID برای زاویه و ارتفاع

پس از تیون کردن ضرایب، رفتار پرنده برای زوایای خواسته شده مسئله و ارتفاع دلخواه 20 متر، به شکل زیر است.



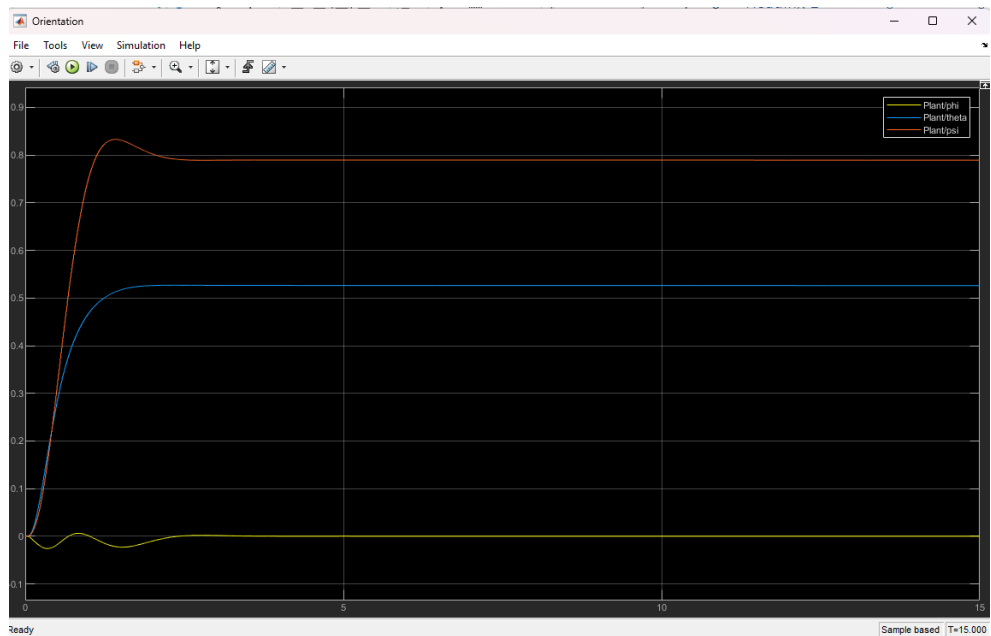


Figure 17: رفتار کنترلر ارتفاع و زاویه

همانطور که مشاهده می‌شود، پرنده دقیقاً به زوایای خواسته شده بر حسب رادیان می‌رسد. همچنین در کمتر از 4 ثانیه از ارتفاع، 10 به ارتفاع 20 متر می‌رود و در همانجا ساکن می‌شود. اما چون سیستم underactuated است، نمیتواند با این کنترلر موقعیت  $x, y$  را حفظ کند مگر آنکه جهت پرنده تغییر کند. همچنین زاویه گاما بر حسب زمان به شکل زیر است. (نقاط ابتدایی که ترسیم نشده به معنای گاما برابر با صفر است.)

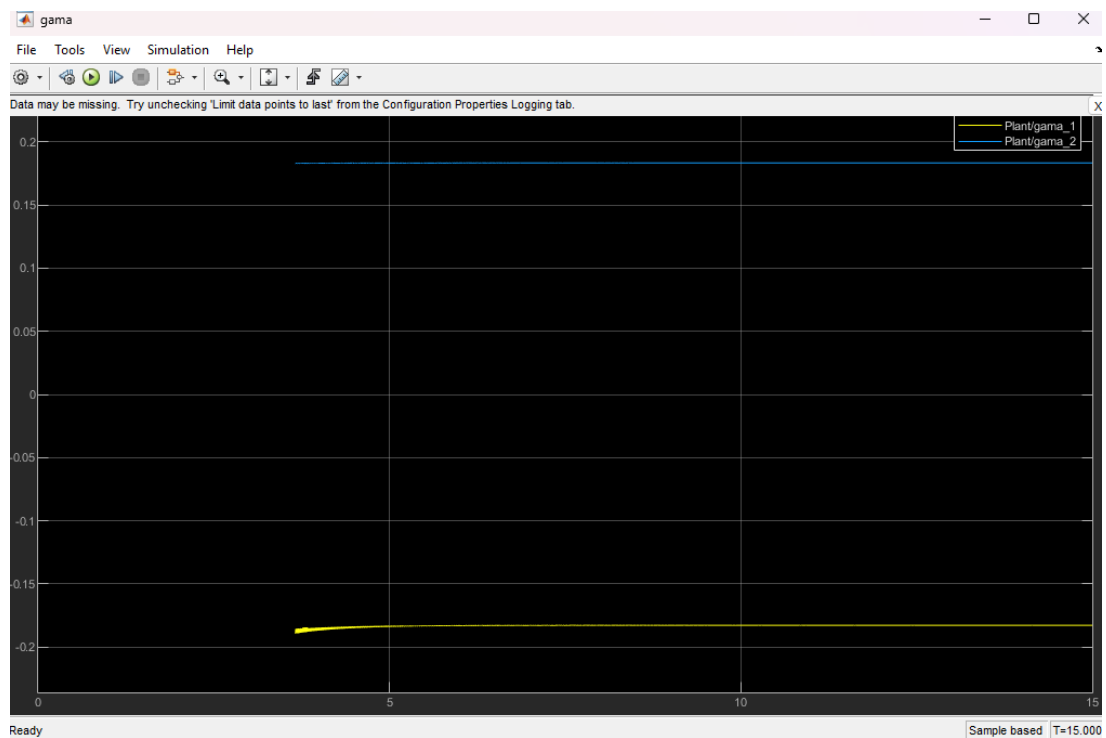


Figure 18: گاما برای کنترلر ارتفاع و زاویه

برای کنترل مکان باید زوایای دوران را طوری تنظیم کنیم که، در موقعیت دلخواه قرار بگیریم. برای این منظور از کنترلی با ساختار cascade استفاده می‌کنیم. (model\_2)

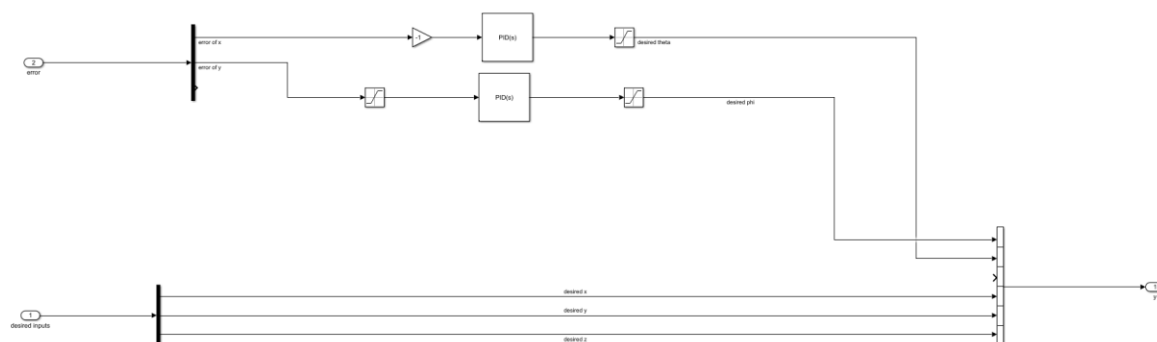
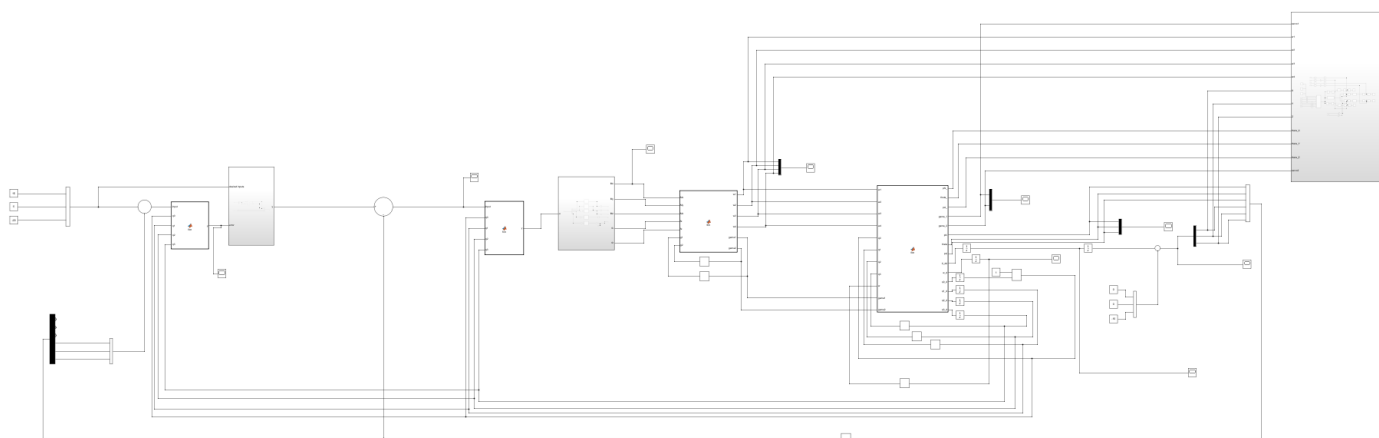
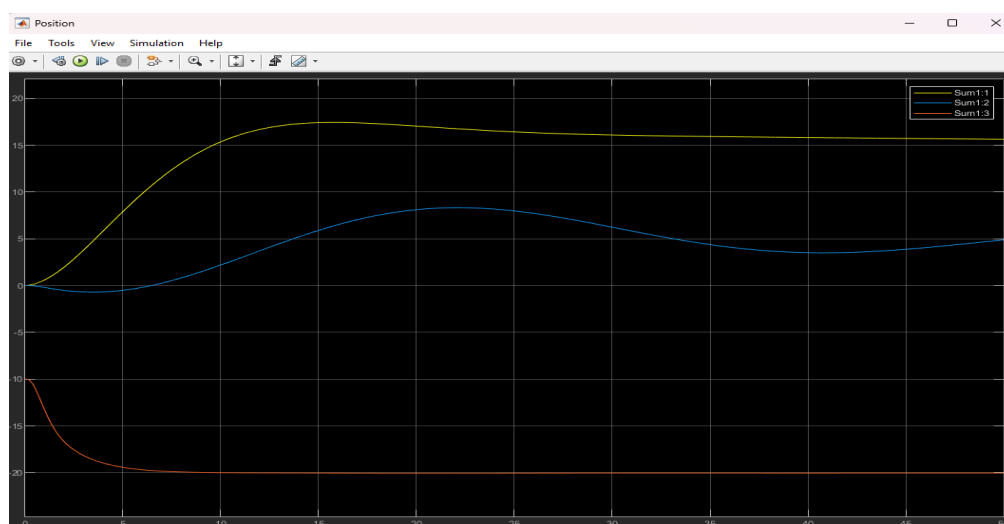


Figure 19: ساختار سیستم و ساختار کنترلر موقعیت

همچنین چون در کنترلر زوایا و ارتفاع که کنترلر داخلی است، نیاز به ورودی‌های اصلی هست، آن‌ها را نیز مستقیم منتقل می‌کنیم. همچنین مانند کنترلر زوایا و ارتفاع، خطای  $x, y$  را باید در دستگاه بدنه دریافت کنیم و به PID بدهیم. سپس کنترلر‌ها را تیون می‌کنیم. در نهایت برای ورودی‌های موقعیت دلخواه (15,5,-20) رفتار سیستم به شکل زیر خواهد بود.



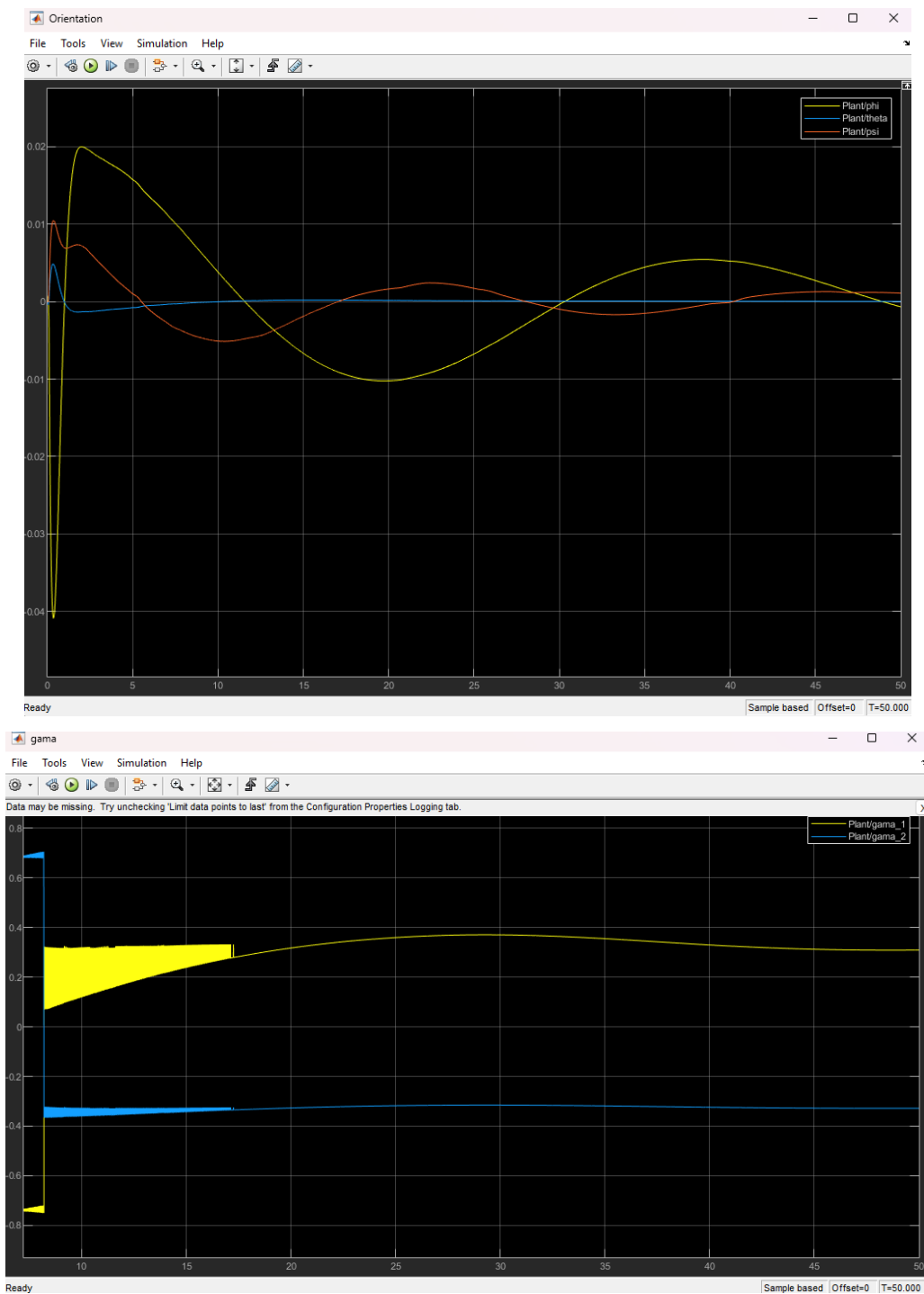


Figure 20: پاسخ کنترلر مکان

همانطور که مشاهده می‌شود، در حدود پرنده نسبتاً سریعتر به موقعیت مورد نظر در  $X, Z$  می‌رسد اما در جهت  $Y$  کنترلر کافی را ندارد. دلیل این مسئله می‌تواند این باشد که دو روتور بالایی، با قابلیت tilt کنترل بهتری بر موقعیت پرنده در جهت  $X$  آن ایجاد می‌کند. چرا که اگر خطای موقعیت در دستگاه بدنه را رسم کنیم، در جهت  $Y$  بدنه کمترین زمان نشست را داریم.

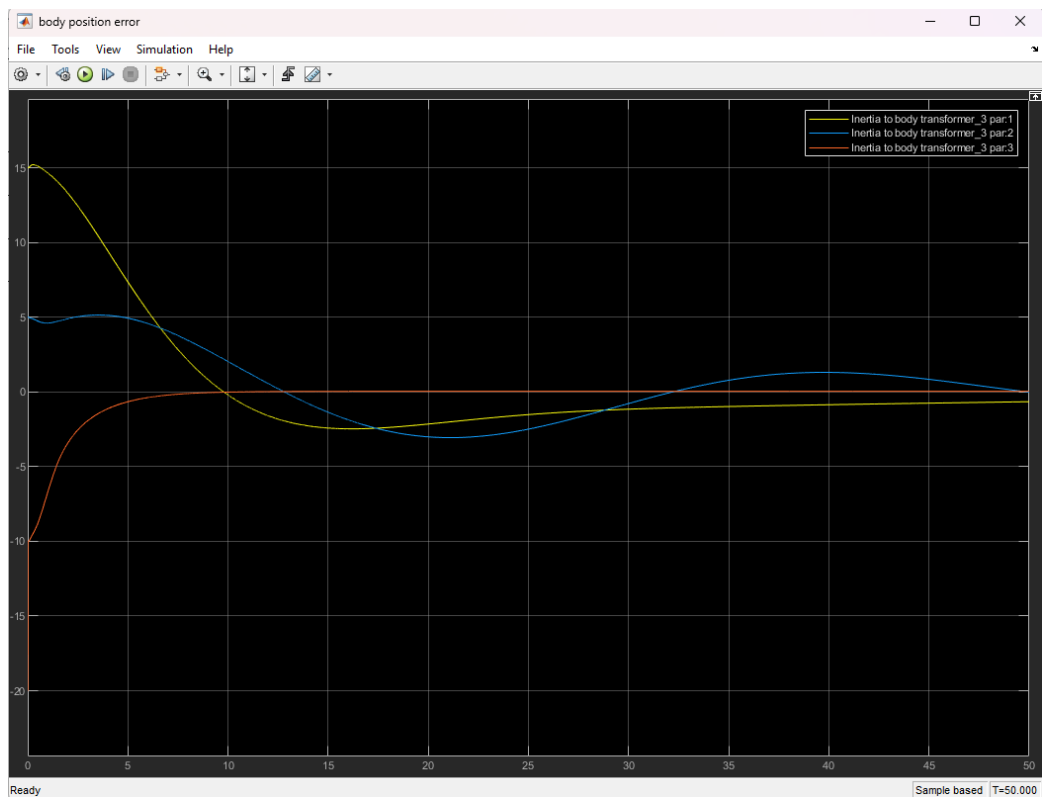


Figure 21: خطای موقعیت در دستگاه بدنه