

دانشگاه صنعتی خواجه نصیرالدین طوسی  
دانشکده مهندسی برق - کروه مهندسی کنترل

## درس کنترل مدرن پروژه پایانی

نام و نام خانوادگی	محمد مهدی کرمی
شماره دانشجویی	۴۰۰۰۸۳۷۳
تاریخ	۱۴۰۳ تیرماه



## فهرست مطالب

۳	سوال یک	۱	
۳	.....	۱.۱ معرفی سیستم	
۳	.....	۲.۱ کاربردها	
۴	.....	۳.۱ اجزای کلیدی	
۴	.....	۴.۱ نمودار بلوکی	
۵		سوال دو	۲
۹		سوال سه	۳
۱۶		سوال چهار	۴
۲۰		سوال پنج	۵



## فهرست تصاویر

۳	کوادکوپتر یکپارچه با دوربین [۲]	۱
۴	نمودار بلوری کوادکوپتر [۴]	۲
۵	نمایش شماتیک کوادکوپتر [۵]	۳
۱۱	سیستم خطی	۴
۱۱	ورودی و خروجی سیستم خطی برای ورودی اول	۵
۱۱	ورودی و خروجی سیستم خطی برای ورودی دوم	۶
۱۲	سیستم غیرخطی	۷
۱۲	ورودی و خروجی سیستم غیرخطی برای ورودی اول	۸
۱۲	ورودی و خروجی سیستم غیرخطی برای ورودی دوم	۹
۱۵	ورودی و خروجی های سیستم خطی	۱۰

## ۱ سوال یک

کوادکوپترها، با ساختاری انعطاف‌پذیر و استفاده از چهار موتور، به عنوان ابزارهای پروازی چند منظوره، در صنایع مختلفی مورد استفاده قرار می‌گیرند. این وسیله پروازی، به علت قابلیت تنظیم و کنترل بالاتر نسبت به سایر وسایل پروازی، در برخی از کاربردهای حیاتی مانند عملیات اضطراری و امنیتی، نقش بسیار مهمی ایفا می‌کند.

### ۱.۱ معرفی سیستم

کوادکوپتر (شکل ۱) که همچنین با عنوان پرنده هدایت‌پذیر از دور (پهپاد) نیز شناخته می‌شود، یک بالگرد چند موتور است که به کمک ۴ موتور بلند شده و به حرکت در می‌آید. همچنین آن در ابعاد و اندازه‌های مختلف وجود دارد. در سال‌های اخیر، محققان نشان دادند که کنترل کوادکوپتر به وسیله روش‌های کنترل خطی با خطی سازی دینامیک‌ها حول نقطه کار امکان‌پذیر است. با این حال، با استفاده از تکنیک‌های کنترل غیرخطی که شکل کلی‌تری از دینامیک‌های کوادکوپتر در تمام مناطق پرواز در نظر گرفته می‌شود، می‌توان به پرواز گستره‌تر و عملکرد بهتر دست یافت [۱].



شکل ۱: کوادکوپتر یکپارچه با دوربین [۲]

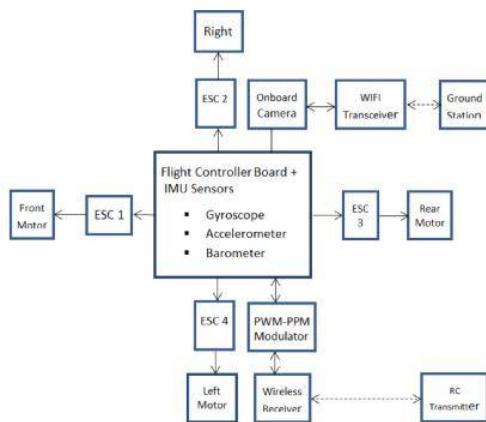
### ۲.۱ کاربردها

کوادکوپترها به عنوان ابزارهای چندمنظوره در محیط‌های مختلف و برای کاربردهای متنوعی استفاده می‌شوند. این ابزارها در عملیات اضطراری مانند پاسخ به حوادث و طوفان‌های طبیعی، شناسایی مناطق آلوده و مانیتورینگ محیطی از اهمیت بسیاری برخوردارند. علاوه بر این، در کاربردهای تجاری، کوادکوپترها برای اسپری کردن محصولات کشاورزی، تحویل کالاها در مناطق شهری، فیلم‌برداری هوایی، مانیتورینگ ساخت و ساز، و حتی تقریحی مورد استفاده قرار می‌گیرند. در کنار اینها، کوادکوپترها در کاربردهای نظامی و امنیتی نیز کارآئی دارند؛ از نظارت بر تجمعات عمومی و پیشگیری از جرم و جنایت گرفته تا استفاده در عملیات نظامی و نظارت بر مرزها و محیط‌های حساس، از جمله کاربردهای مهم آنهاست [۲].

### ۳.۱ اجزای کلیدی

بخش اصلی کوادکوپتر، چارچوب باشد که دارای چهار سبک و سخت باشد تا بتواند یک باتری لیتیوم پلیمر (LIPO)، چهار موتور براشلس DC (BLDC)، برد کنترل، چهار پره، دوربین و انواع مختلف حسگرهای را همراه با یک چارچوب سبک پذیرش کند. برای موقعیت یابی ربات می‌توان از حسگرهای مختلفی مانند سیستم ضبط حرکت (Motion Capture)، واحد اندازه‌گیری (IMU) و سیستم موقعیت یاب جهانی (GPS) استفاده کرد. سرعت موتورهای BLDC توسط کنترل‌کننده سرعت الکترونیکی (ESC) تنظیم می‌شود. برای استحکام بالاتر، یعنی داشتن یک مرکز جرم پایین‌تر، باتری‌ها در نصف پایین قرار داده می‌شوند. موتورها به طور متقاضی از مرکز در دو طرف مخالف قرار دارند. برای جلوگیری از هرگونه تداخل هواپیماگری بین پره‌های پره‌گرد، فاصله بین موتورها به طور تقریبی تنظیم می‌شود. تمام این اجزا بر روی چارچوب یا شاسی اصلی کوادکوپتر نصب می‌شوند [۳].

### ۴.۱ نمودار بلوكى



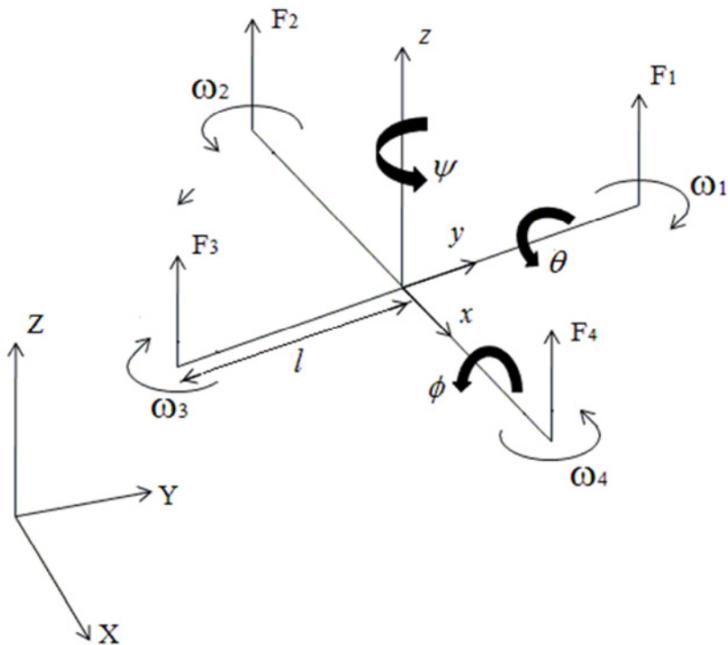
شکل ۲: نمودار بلوكى کوادکوپتر [۴]

شکل ۲ نمایش‌دهنده نمودار بلوك سیستم است. پروژه به چهار بخش اصلی تقسیم می‌شود که شامل واحد ورودی، واحد کنترل، واحد خروجی و واحد تامین و توزیع برق است. در واحد ورودی، یک فرستنده و گیرنده کنترل رادیویی (RC) وجود دارد که برای ارسال کنترل‌ها به کنترل‌کننده پرواز در کوادکوپتر استفاده می‌شود. در واحد کنترل، یک کنترل کننده پرواز به عنوان یک واحد میکروکنترلری ویژه برای تفسیر کنترل‌های RC، ارائه داده‌های تلمتری به یک ایستگاه پایه، و ارائه بازخورد کنترل پویا برای حفظ پایداری کوادکوپتر در حین پرواز استفاده می‌شود. واحد خروجی شامل کنترل‌کننده‌های سرعت الکترونیکی و موتورهای براشلس است. این واحد برای تغییر سرعت موتورها و ارائه ورودی‌های کاربر مختلف کمک می‌کند. در نهایت، واحد تامین و توزیع برق باید منبع تغذیه‌ای باشد که بتواند به راحتی توسط بار موتورها حمل شود و در عین حال به اندازه کافی شارژ برای ESC و موتورها فراهم کند، در حالی که واحد توزیع برق باید بتواند برق مساوی را به چهار ESC و موتور بدون تأخیری که ممکن است خط‌نناک باشد، ارائه دهد [۴].



## ۲ سوال دو

در این بخش، مدل ریاضی کوادکوپتر با در نظر گرفتن سینماتیک و دینامیک آن مورد بررسی قرار می‌گیرد. شکل ۳ نمایی شماتیک از کوادکوپتر را نشان می‌دهد.



شکل ۳: نمایش شماتیک کوادکوپتر [۵]

در این شکل، چهار بازوی کوادکوپتر به طور متقابله قرار گرفته‌اند و در انتهای هر بازو یک موتور DC برای چرخش پروانه نصب شده است. نیروهای رانش به سمت بالا با  $F_1$ ,  $F_2$ ,  $F_3$  و  $F_4$  نشان داده شده‌اند که به ترتیب توسط سرعت‌های زاویه‌ای  $\omega_1$ ,  $\omega_2$ ,  $\omega_3$  و  $\omega_4$  و  $\psi$  نمایش موتورهای DC ایجاد می‌شوند. همچنین، زوایا roll (رول)، pitch (پیچ) و yaw (یاو) در مورد محورهای  $x$ ,  $y$  و  $z$  با  $\theta$ ,  $\phi$  و  $\psi$  نمایش داده شده‌اند. طول بازوی فریم کوادکوپتر با  $l$  مشخص شده است. در شکل ۳، قاب ثابت بدنه با  $z-y-x$  نشان داده شده است، در حالی که قاب مرجع با  $Z-Y-X$  نشان داده می‌شود. تبدیل بین این دو قاب با ماتریس تبدیل  $R$  نشان داده می‌شود که با استفاده از زوایای اویلر ( $\phi$ ,  $\theta$  و  $\psi$ ) در مورد محورهای  $x$ ,  $y$  و  $z$  بدست می‌آید.

$$R = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta c\psi & s\phi s\theta s\psi - c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad (1)$$

در اینجا،  $s\theta = \sin \theta$  و  $c\theta = \cos \theta$ . برای مطالعه‌ی موارد بیشتر درباره‌ی شکل ۳،  $\omega_1$  و  $\omega_3$  در جهت عقربه‌های ساعت و  $\omega_2$  و  $\omega_4$  در جهت خلاف عقربه‌های ساعت می‌چرخند تا گشتاور حول محور  $z$  متعادل شود. تفاوت بین  $\omega_1$  و  $\omega_3$  و گشتاور  $T_x$  حول محور  $x$  صورت زیر است.

$$T_x = (\omega_1^2 - \omega_3^2)K_t l \quad (2)$$

همچنین، تفاوت بین  $\omega_2$  و  $\omega_4$  و گشتاور  $T_y$  حول محور  $y$  به صورت زیر است.

$$T_y = (\omega_2^2 - \omega_4^2)K_t l \quad (3)$$

به طور مشابه، حرکت انحراف زمانی به دست می‌آید که در سرعت‌های زاویه‌ای دو جفت پروانه تفاوت وجود داشته باشد. این انحراف زمانی با  $T_z$  که گشتاور حول محور  $z$  است، نشان داده می‌شود و به صورت زیر محاسبه می‌شود.

$$T_z = (\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2)K_d \quad (4)$$

در اینجا،  $K_d$  و  $K_t$  ضرایبی هستند که به چگالی هوا، هندسه پروانه و شعاع پره بستگی دارند. بر اساس معادلات اویلر، گشتاورهای مربوط به محورهای  $x$ ,  $y$  و  $z$  با معادله زیر نشان داده می‌شوند.

$$\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} I_x \ddot{\phi} \\ I_y \ddot{\theta} \\ I_z \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_x \dot{\phi} \\ I_y \dot{\theta} \\ I_z \dot{\psi} \end{bmatrix} \quad (5)$$

بر اساس معادله نیوتن، معادله حرکت به صورت زیر است.

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = R \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (6)$$

در اینجا،  $F = F_1 + F_2 + F_3 + F_4$  و  $g$  به ترتیب جرم کوادکوپتر و شتاب ناشری از گرانش را نشان می‌دهند. با استفاده از معادله ۶ تا ۴ معادله حرکت کوادکوپتر به دست می‌آید.

$$m\ddot{x} = (F_1 + F_2 + F_3 + F_4)(c\phi s\theta c\psi + s\phi s\psi) \quad (7)$$

$$m\ddot{y} = (F_1 + F_2 + F_3 + F_4)(c\phi s\theta s\psi + s\phi c\psi) \quad (8)$$

$$m\ddot{z} = (F_1 + F_2 + F_3 + F_4)(c\phi c\theta) - mg \quad (9)$$

$$I_x \ddot{\phi} = (F_1 - F_3)l + \dot{\theta} \dot{\psi}(I_y - I_z) \quad (10)$$

$$I_y \ddot{\theta} = (F_2 - F_4)l + \dot{\psi} \dot{\phi}(I_z - I_x) \quad (11)$$

$$I_z \ddot{\psi} = (M_2 + M_4 - M_1 - M_3)l + \dot{\phi} \dot{\theta}(I_x - I_y) \quad (12)$$

این معادلات نشان دهنده مدل دینامیکی غیرخطی کوادکوپتر هستند. مدل در مورد نقطه‌ی شناور تعادل خطی شده است و سیستم خطی معادله به صورت زیر آورده شده است.

$$\ddot{x} = \frac{F}{m}\theta \quad (13)$$

$$\ddot{y} = \frac{F}{m}\phi \quad (14)$$

$$\ddot{z} = \frac{F}{m} - g \quad (15)$$

$$\ddot{\phi} = \frac{T_x}{I_x} \quad (16)$$

$$\ddot{\theta} = \frac{T_y}{I_y} \quad (17)$$

$$\ddot{\psi} = \frac{T_z}{I_z} \quad (18)$$



حال می خواهیم مدل را به شکل فضای حالت بازنویسی کنیم.

$$\begin{aligned}
 x_1 &= x & x_2 &= \dot{x}_1 = \dot{x} \\
 x_3 &= y & x_4 &= \dot{x}_3 = \dot{y} \\
 x_5 &= z & x_6 &= \dot{x}_5 = \dot{z} \\
 x_7 &= \phi & x_8 &= \dot{x}_7 = \dot{\phi} \\
 x_9 &= \theta & x_{10} &= \dot{x}_9 = \dot{\theta} \\
 x_{11} &= \psi & x_{12} &= \dot{x}_{11} = \dot{\psi}
 \end{aligned} \tag{۱۹}$$

حال، مدل را به شکل فضای حالت بازنویسی می کنیم:

$$\begin{aligned}
 \dot{X} &= AX + BU \\
 Y &= CX + DU
 \end{aligned} \tag{۲۰}$$

در این معادله،  $X$ ،  $U$  و  $Y$  به ترتیب بردارهای حالت، ورودی و خروجی هستند. معادلات حرکت و خروجی با ماتریس‌های  $A$ ،  $B$  و  $C$  نشان داده می‌شوند. در اینجا  $D$  یک ماتریس خروجی صفر است.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix} \begin{bmatrix} F \\ T_x \\ T_y \\ T_z \end{bmatrix} \tag{۲۱}$$



$$\begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} \quad (22)$$

این معادلات به شکل فضایی حالت برای دینامیک کوادکوپتر نوشته شده‌اند.

به دلیل سخت بودن کنترل و مدلسازی چنین سیستمی، سعی می‌کنیم که کوادکوپتر را در یک فضای دو بعدی  $Y - Z$  بررسی کنیم.  $y, z$  و  $\phi$  را به عنوان متغیرهای مورد نظر برای بررسی و کنترل درنظر می‌گیریم. دو ورودی  $u_1$  و  $u_2$ ، ورودی‌های سیستم خواهند بود. معادلات غیرخطی به صورت زیر می‌باشند:

$$\begin{aligned} \ddot{y} &= -\frac{u_1}{m} \sin \phi \\ \ddot{z} &= -g + \frac{u_1}{m} \cos \phi \\ \ddot{\phi} &= \frac{u_2}{I_x} \end{aligned} \quad (23)$$

در صورت خطی سازی، حول نقطه تعادل، معادلات به صورت زیر می‌باشند:

$$\begin{aligned} \dot{y} &= -g\phi \\ \dot{z} &= -g + \frac{u_1}{m} \\ \dot{\phi} &= \frac{u_2}{I_x} \end{aligned} \quad (24)$$

برای طراحی کنترل برای  $y$  باید درنظر داشت که  $\phi$  به عنوان یک لوب داخلی عمل می‌کند و باید سریع تر از  $z$  باشد. معادلات را به شکل فضایی حالت می‌نویسیم؛ ماتریس‌های  $A, B, C$  و  $D$  به شکل زیر خواهند بود:

$$\begin{bmatrix} \dot{y} \\ \dot{z} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -g \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ z \\ \phi \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & \frac{1}{I_x} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (25)$$



$$y = \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ z \\ \phi \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (26)$$

### سوال سه ۳

با کمک MATLAB، ماتریس انتقال حالت را محاسبه می‌کنیم:

```

1 % Constants
2 g = 9.81;
3 i_x = 4.85e-3;
4
5 % Define the state-space matrices A, B, C, D
6 A = [0 0 0 0 0 -g;
7     0 0 0 0 0 0;
8     0 0 0 0 0 0;
9     1 0 0 0 0 0;
10    0 1 0 0 0 0;
11    0 0 1 0 0 0];
12
13 B = [0 0;
14     1 0;
15     0 1/i_x;
16     0 0;
17     0 0;
18     0 0];
19
20 C = [0 0 0 0 1 0;
21     0 0 0 1 0 0];
22
23 D = [0 0;
24     0 0];
25
26 % Display the state-space matrices

```



```

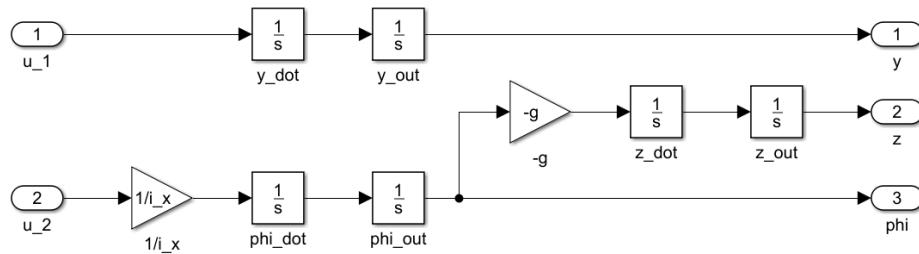
27 disp('State-Space Matrices:');
28 disp('A:');
29 disp(A);
30 disp('B:');
31 disp(B);
32 disp('C:');
33 disp(C);
34 disp('D:');
35 disp(D);
36
37 % Create the state-space system
38 sys = ss(A, B, C, D);
39
40 % Display the state-space system
41 disp('State-Space System:');
42 disp(sys);
43
44 % Define the time variable t
45 t = sym('t');
46
47 % Calculate the state transition matrix for a general time t
48 Phi_t = expm(A * t);
49
50 % Display the state transition matrix
51 disp('State Transition Matrix for general time t:');
52 disp(Phi_t);

```

خروجی کد بالا ماتریس زیر است:

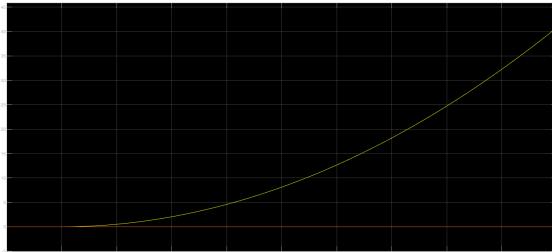
$$\Phi(t) = \begin{bmatrix} 1 & 0 & -\frac{981t^2}{200} & 0 & 0 & -\frac{981t}{100} \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ t & 0 & -\frac{327t^3}{200} & 1 & 0 & -\frac{981t^2}{200} \\ 0 & t & 0 & 0 & 1 & 0 \\ 0 & 0 & t & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

مدل خطی را در سیمولینک پیاده سازی می کنیم؛ دوروری مختلف به سیستم خطی می دهیم و خروجی را به ازای ورودی های مختلف مشاهده می کنیم.

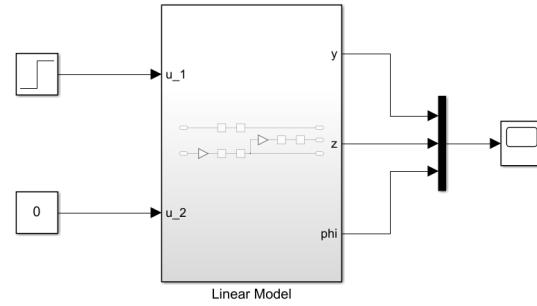


شکل ۴: سیستم خطی

در ورودی اول، به  $u_1$  یک تابع پله واحد و به  $u_2$  یک تابع ثابت صفر را می‌دهیم.



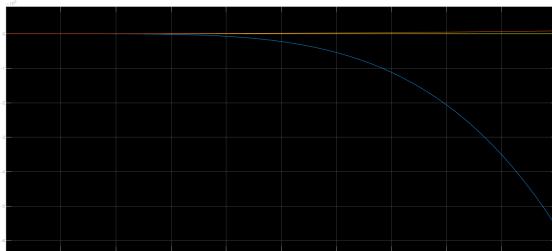
(ب) خروجی اول سیستم خطی



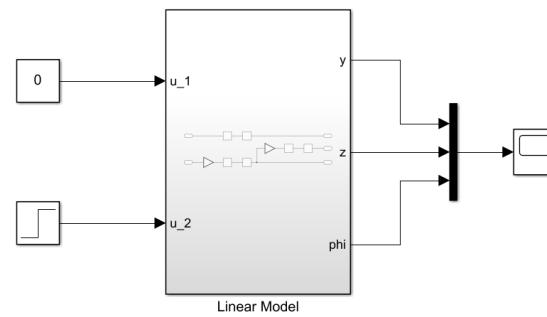
(ا) ورودی اول سیستم خطی

شکل ۵: ورودی و خروجی سیستم خطی برای ورودی اول

در ورودی دوم، به  $u_1$  یک تابع ثابت صفر و به  $u_2$  یک تابع پله واحد را می‌دهیم.



(ب) خروجی دوم سیستم خطی

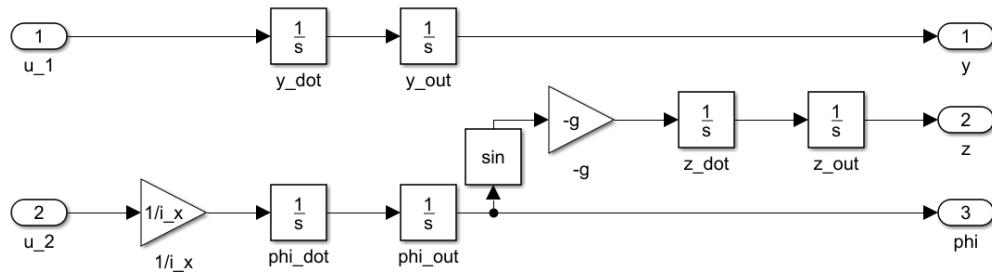


(ا) ورودی دوم سیستم خطی

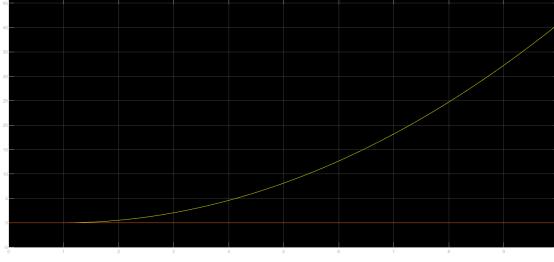
شکل ۶: ورودی و خروجی سیستم خطی برای ورودی دوم

سپس مدل غیرخطی را در سیمولینک پیاده سازی می‌کنیم؛ دو وروری مختلف به سیستم غیرخطی می‌دهیم و خروجی را به ازای ورودی‌های مختلف مشاهده می‌کنیم.

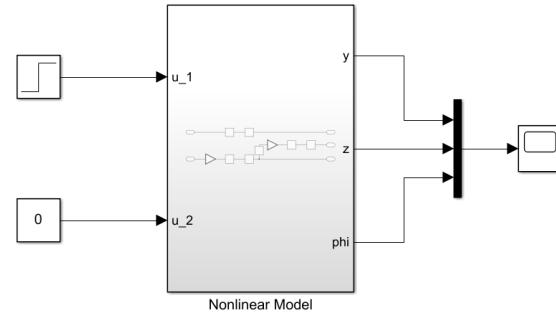
در ورودی اول، به  $u_1$  یک تابع پله واحد و به  $u_2$  یک تابع ثابت صفر را می‌دهیم.



شکل ۷: سیستم غیرخطی



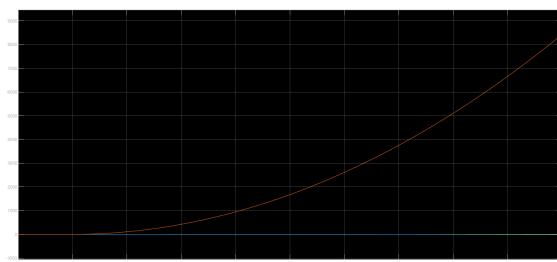
(ب) خروجی اول سیستم غیرخطی



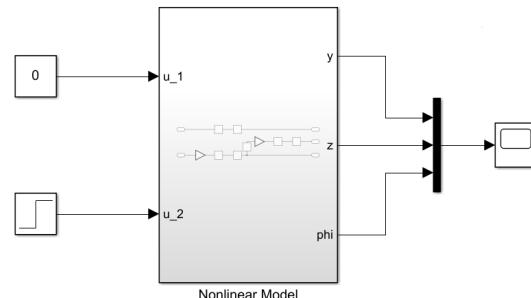
(ا) ورودی اول سیستم غیرخطی

شکل ۸: ورودی و خروجی سیستم غیرخطی برای ورودی اول

در ورودی دوم، به  $u_1$  یک تابع ثابت صفر و به  $u_2$  یک تابع پله واحد را می‌دهیم.



(ب) خروجی دوم سیستم غیرخطی



(ا) ورودی دوم سیستم غیرخطی

شکل ۹: ورودی و خروجی سیستم غیرخطی برای ورودی دوم

همچنین با کمک ماتریس‌های فضای حالت، پاسخ مدل خطی را با MATLAB بررسی می‌کنیم.

```

1 % Constants
2 g = 9.81;
3 i_x = 4.85e-3;
```



```

4
5 % Define the state-space matrices A, B, C, D
6 A = [0 0 0 0 0 -g;
7     0 0 0 0 0 0;
8     0 0 0 0 0 0;
9     1 0 0 0 0 0;
10    0 1 0 0 0 0;
11    0 0 1 0 0 0];
12
13 B = [0 0;
14     1 0;
15     0 1/i_x;
16     0 0;
17     0 0;
18     0 0];
19
20 C = [0 0 0 0 1 0;
21     0 0 0 1 0 0];
22
23 D = [0 0;
24     0 0];
25
26 % Create the state-space system
27 sys = ss(A, B, C, D);
28
29 % Time vector
30 t = 0:0.02:10; % Simulation time from 0 to 10 seconds
31
32 % Define input signals
33 u1_in = [ones(size(t)); % Step input for u1
34     zeros(size(t))]; % Zero input for u2
35
36 u2_in = [zeros(size(t)); % Zero input for u1
37     ones(size(t))]; % Step input for u2
38
39 % Simulate the system's response to each input

```



```

40 [y_out_1, ~, ~] = lsim(sys, u1_in, t); % Output response to input u1_in
41 [y_out_2, ~, ~] = lsim(sys, u2_in, t); % Output response to input u2_in
42
43 % Plotting
44 figure;
45
46 % Plot for input signals
47 subplot(2, 2, 1);
48 plot(t, u1_in(1,:), 'b', 'LineWidth', 1.5); % Plot u1
49 hold on;
50 plot(t, u1_in(2,:), 'r', 'LineWidth', 1.5); % Plot u2
51 title('Input 1 Signals');
52 xlabel('Time (seconds)');
53 ylabel('Input');
54 legend('u1', 'u2');
55 grid on;
56
57 subplot(2, 2, 2);
58 plot(t, u2_in(1,:), 'b', 'LineWidth', 1.5); % Plot u1
59 hold on;
60 plot(t, u2_in(2,:), 'r', 'LineWidth', 1.5); % Plot u2
61 title('Input 2 Signals');
62 xlabel('Time (seconds)');
63 ylabel('Input');
64 legend('u1', 'u2');
65 grid on;
66
67 % Plot for output signals
68 subplot(2, 2, 3);
69 plot(t, y_out_1(:,1), 'b', 'LineWidth', 1.5); % Output y to u1
70 hold on;
71 plot(t, y_out_1(:,2), 'r', 'LineWidth', 1.5); % Output z to u1
72 title('Output Signals for Input 1');
73 xlabel('Time (seconds)');
74 ylabel('Output');
75 legend('y', 'z');

```

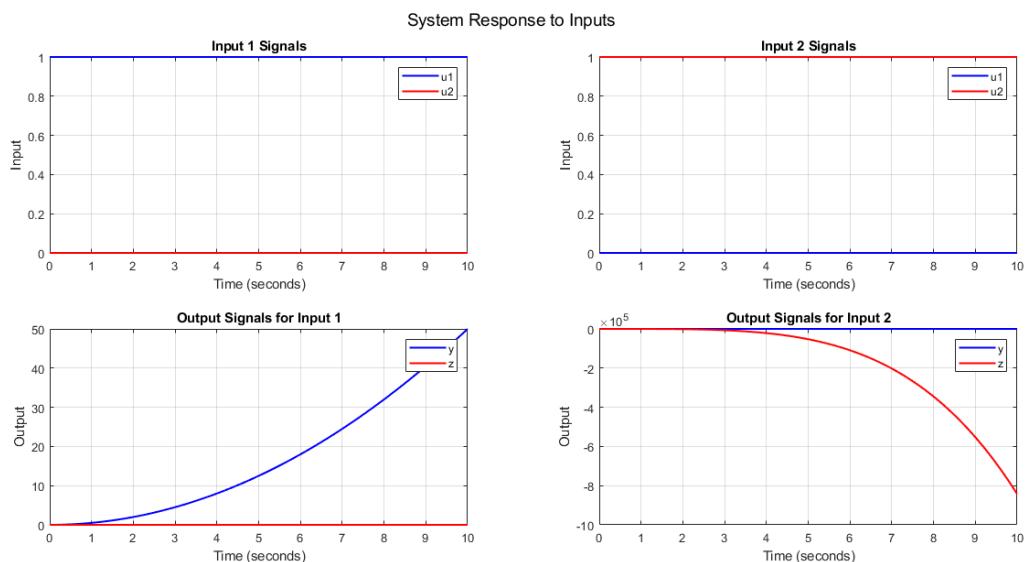


```

76 grid on;
77
78 subplot(2, 2, 4);
79 plot(t, y_out_2(:,1), 'b', 'LineWidth', 1.5); % Output y to u2
80 hold on;
81 plot(t, y_out_2(:,2), 'r', 'LineWidth', 1.5); % Output z to u2
82 title('Output Signals for Input 2');
83 xlabel('Time (seconds)');
84 ylabel('Output');
85 legend('y', 'z');
86 grid on;
87
88 % Adjust figure
89 sgttitle('System Response to Inputs');

```

خروجی کد بالا نمودارهای زیر است؛ دو ورودی متفاوت و خروجی‌های آنها.



شکل ۱۰: ورودی و خروجی‌های سیستم خطی



## ۴ سوال چهار

با کمک MATLAB، ماتریس‌های رویت‌پذیری و کنترل‌پذیری را به کمک ماتریس‌های فضای حالت به دست می‌آوریم و رویت‌پذیری و کنترل‌پذیری سیستم را بررسی می‌کنیم.

```

1 % Constants
2 g = 9.81;
3 i_x = 4.85e-3;
4
5 % Define the state-space matrices A, B, C, D
6 A = [0 0 0 0 0 -g;
7     0 0 0 0 0 0;
8     0 0 0 0 0 0;
9     1 0 0 0 0 0;
10    0 1 0 0 0 0;
11    0 0 1 0 0 0];
12
13 B = [0 0;
14     1 0;
15     0 1/i_x;
16     0 0;
17     0 0;
18     0 0];
19
20 C = [0 0 0 0 1 0;
21     0 0 0 1 0 0];
22
23 D = [0 0;
24     0 0];
25
26 % Create the state-space system
27 sys = ss(A, B, C, D);
28
29 % Check observability
30 O = obsv(sys);
31 rank_O = rank(O);

```



```

32
33 % Display results
34 disp('Observability Matrix:');
35 disp(0);
36 fprintf('Rank of Observability Matrix: %d\n', rank_0);
37 if rank_0 == size(A, 1)
38     disp('System is observable.');
39 else
40     disp('System is not observable.');
41 end
42
43 % Check controllability
44 Co = ctrb(sys);
45 rank_Co = rank(Co);
46
47 % Display results
48 disp('Controllability Matrix:');
49 disp(Co);
50 fprintf('Rank of Controllability Matrix: %d\n', rank_Co);
51 if rank_Co == size(A, 1)
52     disp('System is controllable.');
53 else
54     disp('System is not controllable.');
55 end

```

خروجی کد در ادامه نمایش داده شده است؛ ماتریس رویت‌پذیری به شکل زیر است:



$$O = \begin{bmatrix} 0 & 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 & 0 \\ 1.0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -9.8100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -9.8100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (۲۸)$$

رنک ماتریس رویت‌پذیری برابر ۶ می‌باشد؛ پس سیستم رویت‌پذیر است. ماتریس کنترل‌پذیری به شکل زیر است:

$$C = 10^3 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -2.0227 & 0 & 0 & 0 & 0 & 0 \\ 0.0010 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2062 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2.0227 & 0 & 0 & 0 \\ 0 & 0 & 0.0010 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2062 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (۲۹)$$

رنک ماتریس کنترل‌پذیری برابر ۶ می‌باشد؛ پس سیستم کنترل‌پذیر است.

سپس سیستم خود را به فرم قطری‌بلوکی جردن تبدیل می‌کنیم؛ برای این کار از MATLAB کمک می‌گیریم:

```

1 % Constants
2 g = 9.81;
3 i_x = 4.85e-3;
4
5 % Define the state-space matrices A, B, C, D
6 A = [0 0 0 0 0 -g;
7     0 0 0 0 0 0;
8     0 0 0 0 0 0;
9     1 0 0 0 0 0;
10    0 1 0 0 0 0;
11    0 0 1 0 0 0];

```



```

12
13 B = [0 0;
14     1 0;
15     0 1/i_x;
16     0 0;
17     0 0;
18     0 0];
19
20 C = [0 0 0 0 1 0;
21     0 0 0 1 0 0];
22
23 D = [0 0;
24     0 0];
25
26 % Calculate the Jordan form of matrix A
27 [T,J] = jordan(A);
28
29 % Display the Jordan form and the similarity transformation matrix
30 disp('Jordan form of matrix A (J):');
31 disp(J);
32
33 disp('Similarity transformation matrix (T):');
34 disp(T);

```

فرم جردن به شکل زیر مشاهده می شود:

$$J = T^{-1}AT = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (30)$$

ماتریس تبدیل همانند هم به این شکل خواهد بود:



$$T = \begin{bmatrix} 0 & -9.8100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0000 \\ 0 & 0 & 0 & 1.0000 & 0 & 0 \\ -9.8100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0000 \\ 0 & 0 & 1.0000 & 0 & 0 & 0 \end{bmatrix} \quad (31)$$

## ۵ سوال پنج

کد متلب زیر،تابع تبدیل سیستم را از روی فضای حالت به دست می آورد، سپس یک تحقیق برای آن محاسبه می کند.

```

1 % Constants
2 g = 9.81;
3 i_x = 4.85e-3;
4
5 % Define the state-space matrices A, B, C, D
6 A = [0 0 0 0 0 -g;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      1 0 0 0 0 0;
      0 1 0 0 0 0;
      0 0 1 0 0 0];
7
8
9
10
11
12
13 B = [0 0;
      1 0;
      0 1/i_x;
      0 0;
      0 0;
      0 0];
14
15
16
17
18
19
20 C = [0 0 0 0 1 0;
      0 0 0 1 0 0];
21
22
23 D = [0 0;
      0 0];
24

```



```

25
26 % Create state-space model
27 sys = ss(A, B, C, D);
28
29 % Convert to transfer function
30 G = tf(sys);
31
32 % Display transfer functions from input to output
33 disp('Transfer Functions:');
34 [num_outputs, num_inputs] = size(G);
35 for out = 1:num_outputs
36     for in = 1:num_inputs
37         fprintf('From Input %d to Output %d:\n', in, out);
38         G(out, in)
39     end
40 end
41
42 % Determine the number of outputs and inputs
43 [num_outputs, num_inputs] = size(G);
44
45 % Initialize cell arrays to store state-space models
46 sys_ss = cell(num_outputs, num_inputs);
47
48 % Create state-space realizations for each transfer function
49 for out = 1:num_outputs
50     for in = 1:num_inputs
51         % Extract transfer function Gij
52         Gij = G(out, in);
53
54         % Create state-space realization
55         sys_ss{out, in} = ss(Gij);
56     end
57 end
58
59 % Initialize an empty state-space model
60 sys_merged = ss([], [], [], []);

```



```

61
62 % Append all subsystems into a single state-space model
63 for out = 1:num_outputs
64     for in = 1:num_inputs
65         sys_merged = append(sys_merged, sys_ss{out, in});
66     end
67 end
68
69 % Display the state-space matrices A, B, C, D of the model
70 disp('State-Space Model:');
71 sys_merged

```

مشاهده می‌شود که توابع انتقال، از ورودی ۱ به خروجی ۱:  $H_{11}(s) = \frac{1}{s^2}$ ، از ورودی ۲ به خروجی ۱:  $H_{21}(s) = 0$ ؛ از ورودی ۱ به خروجی ۲:  $H_{12}(s) = \frac{-2023}{s^4}$  و از ورودی ۲ به خروجی ۲:  $H_{22}(s) = 0$  هستند.  
تحقیق نیز به شکل زیر به دست آمده است:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 32 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (۳۲)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -63.21 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (۳۳)$$

پایداری داخلی و پایداری ورودی محدود-خروجی محدود را بررسی می‌کنیم.

```

1 % Constants
2 g = 9.81;
3 i_x = 4.85e-3;
4

```



```

5 % Define the state-space matrices A, B, C, D
6 A = [0 0 0 0 0 -g;
7     0 0 0 0 0 0;
8     0 0 0 0 0 0;
9     1 0 0 0 0 0;
10    0 1 0 0 0 0;
11    0 0 1 0 0 0];
12
13 B = [0 0;
14     1 0;
15     0 1/i_x;
16     0 0;
17     0 0;
18     0 0];
19
20 C = [0 0 0 0 1 0;
21     0 0 0 1 0 0];
22
23 D = [0 0;
24     0 0];
25
26 % Check BIBO stability
27 isBIBOStable = isstable(ss(A, B, C, D));
28 fprintf('BIBO Stability: %d\n', isBIBOStable);
29
30 % Compute eigenvalues of matrix A
31 eigenvalues = eig(A);
32
33 % Check internal stability
34 isInternalStable = all(real(eigenvalues) < 0);
35 fprintf('Internal Stability: %d\n', isInternalStable);

```

پس از اجرای کد بالا مشاهده می‌شود که سیستم پایدار نیست.



## منابع

- [1] Sabir Abdelhay and Alia Zakriti. "Modeling of a quadcopter trajectory tracking system using PID controller". In: *Procedia Manufacturing* 32 (2019), pp. 564–571.
- [2] Gordana Ostojić et al. "Design, control and application of quadcopter". In: *International Journal of Industrial Engineering and Management* 6.1 (2015), p. 43.
- [3] Omkar Tatale et al. "Quadcopter: design, construction and testing". In: *International Journal for Research in Engineering Application & Management* 4 (2018), pp. 1–7.
- [4] BO Omijeh et al. "Design and Construction of a Quadcopter with Payload for Pipeline Inspection and Surveillance". In: *Intl. J. of Advanced Research in Electrical, Electronics and Instrumentation Engineering* 5.6 (2016).
- [5] Faraz Ahmad et al. "Simulation of the Quadcopter Dynamics with LQR based Control". In: *Materials Today: Proceedings* 24 (2020), pp. 326–332.