کسری حسنی 9923107

گزارشکار Anti windup

در پوشه PID Anti-Windup دو پوشه وجود دارد به نام های 1 و 2

در پوشه 1 پیاده سازی PID در script صورت گرفته است که برای نشان دادن نمودار های PID با امگا های مختلف هست.

در پوشه 2 پیاده سازی Anti Windup به دو روش script وSimulink انجام شده است که در script با تغییر Ta ها و رسم نمودار های حالت های مختلف خروجی تاثیر افزایش و کاهش ثابت زمانی Ta که از بازه 1 تا 10 هست بر روی ناحیه اشباع نشان داده شده است.

در ادامه به توضیح هر کدام می پردازیم:

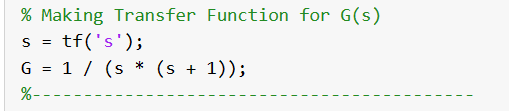
**بخش اول:**

پاک سازی و شروع دوباره و بستن پروژه ها و کار های قبلی در متلب:

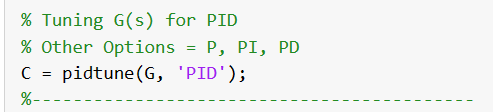
A close-up of a white background

Description automatically generated

ساخت تابع تبدیل G(s) = 1 / s (s+1):

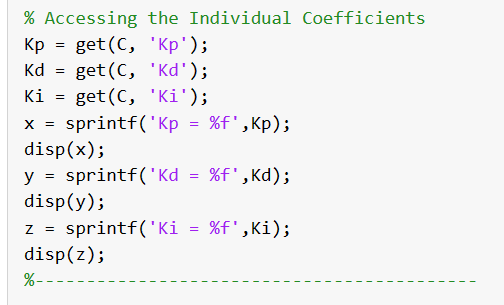


ست کردن یا تیون کردن پارامتر های که PID (Kp، Kd و Ki) با استفاده از تابع pidtune برای تابع G(s):



از این تابع pidtune می توان برای ست کردن پارامتر ها برای هر 4 روش P، PI، PD و PID استفاده کرد و فقط کافیست mode آن را تغییر داد.

در بخش بعدی کد ضرایب PID (Kp، Kd و Ki) در خروجی نمایش داده شده اند:



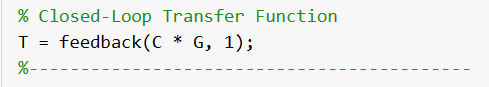
که مقادیر خروجی به شرح زیر می باشد:

Kp = 2.109126

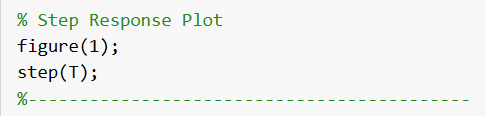
Kd = 1.834131

Ki = 0.538966

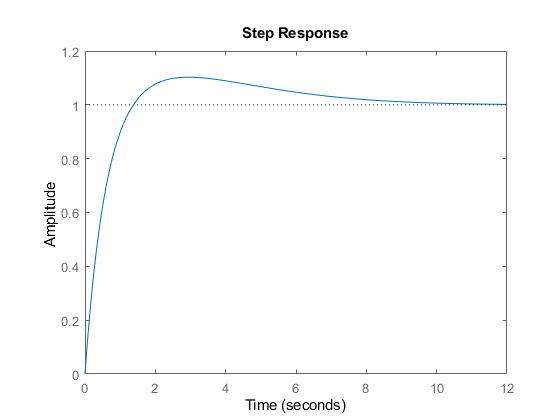
در ادامه سیستم PID را به صورت حلقه بسته با فیدبک 1 می سازیم که در نهایت تابع تبدیل T می شود:



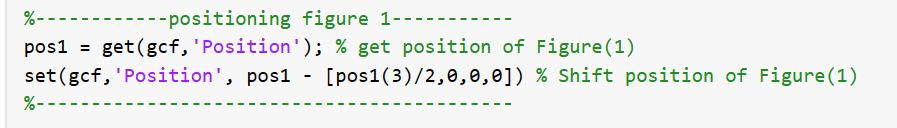
در مرحله بعدی کد نمودار پاسخ پله تابع تبدیل T را با دستور زیر رسم می کنیم:



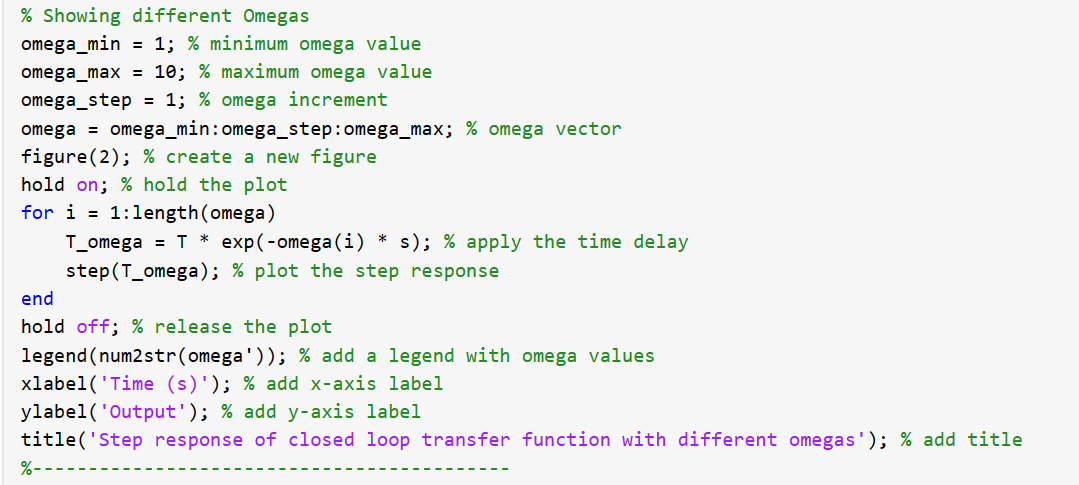
نمودار آن نیز به شکل زیر می شود:



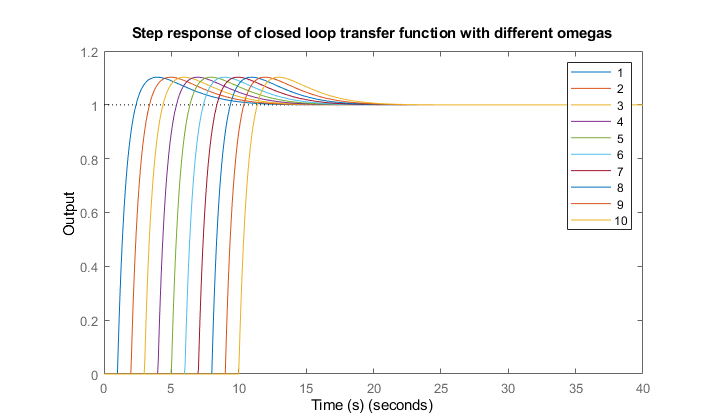
در بخش بعدی کد figure 1 که همان نمودار پاسخ پله تابع تبدیل T هست مختصاتی را اخذ می کند تا بتوان figure 2 را در کنار آن در قسمت های بعدی کد به صورت دو پنجره کنار هم گذاشت:



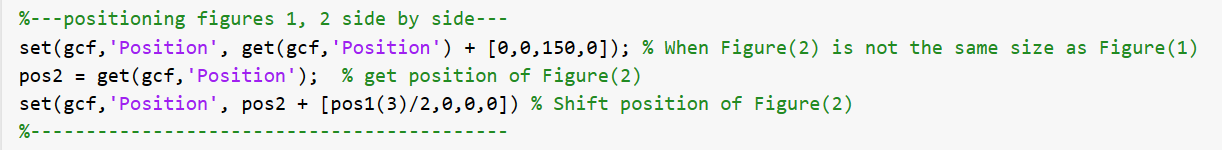
در بخش بعدی کد نمودار پاسخ پله تابع تبدیل T را به ازای یعنی از 1 تا 10 با گام های یک رسم می کنیم:



*نمودار حاصله رسم شده مطابق شکل زیر می باشد:*

**

*در نهایت هم figure 1 و figure 2 در کنار همدیگه قرار داده شده اند:*

**

*شما می توانید برای اطلاعات دقیق تر و نحوه عملکرد کد نوشته شده فایل html پابلیش شده توسط خود برنامه متلب را در همین پوشه 1 ملاحظه بفرمایید.*

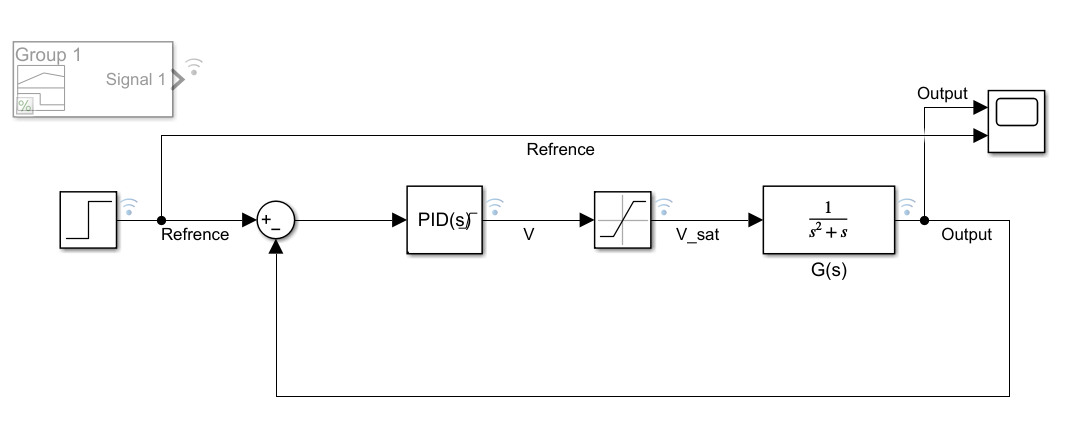
***بخش دوم:***

*در این بخش ابتدا یک فایل سیمولینگ هست که شبیه سازی در آن برای Kb = 1 / Ta ثابت انجام شده است و شما می توانید نتیجه را هم در اسکوپ و هم توسط inspector مشاهده نمایید.*

*ابتدا در مورد شماتیک و بلوک دیاگرام ساخته شده و نحوه پیاده سازی آنتی وینداپ در سیمولینک توضیح مختصری می دهیم و در ادامه کد k2.m مورد بررسی قرار می گیرد که در آن کنترلر pid با روش آنتی وینداپ پیاده سازی شده و با تغییر مقادیر Ta رفتار تابع نهایی در ناحیه اشباع بررسی می شود.*

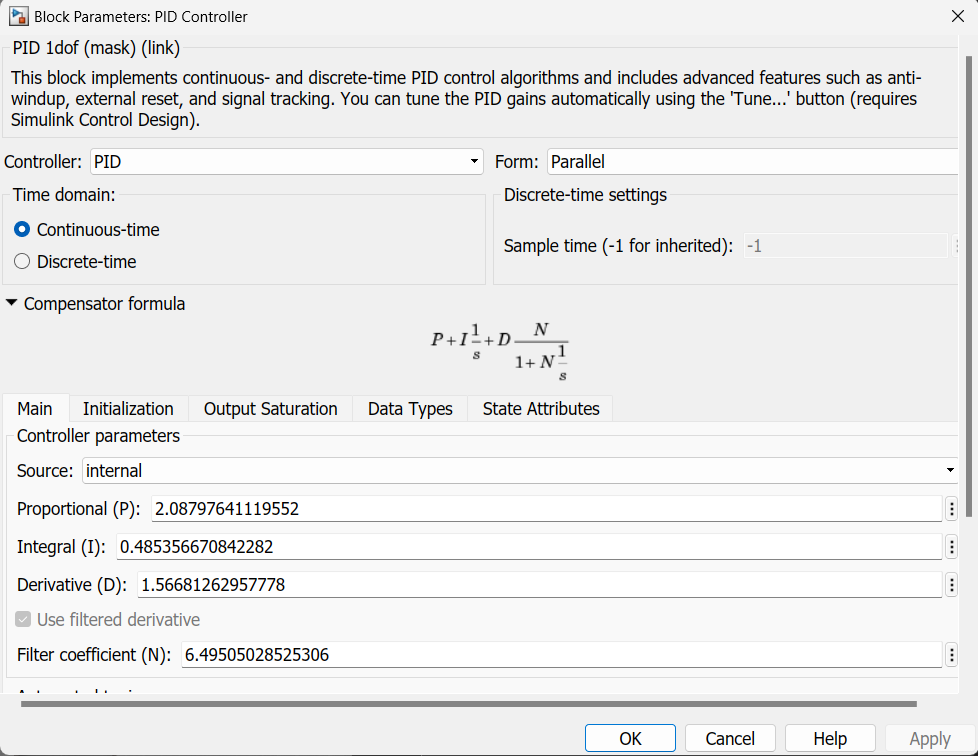
***بخش Simulink:***

*بلوک دیاگرام آن به شکل زیر است:*

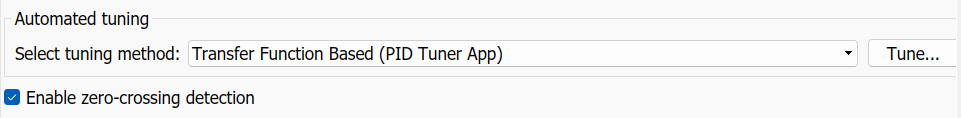
**

*در اینجا در ورودی تابع پله را داریم که بعد به یک جمع کننده می رود و خروجی جمع کننده به سمت PID رفته در نهایت خروجی PID را دچار saturation می کنیم و آن را به تابع تبدیل G(s) که داشتیم تحویل می دهیم.*

*در بخش طراحی این بلوک دیاگرام در تنظیمات PID Controller داریم:*

**

*که این مقادیر P، I، D همان مقادیر* Kp، Kd و Ki در تابع تبدیل PID هستند که با استفاده از خاصیت Automated tuning خود Simulink انجام شده است که با مقایسه کردن با مقادیر برنامه اول می توانید ببینید که پارامتر های PID به همدیگر نزدیک هستند:

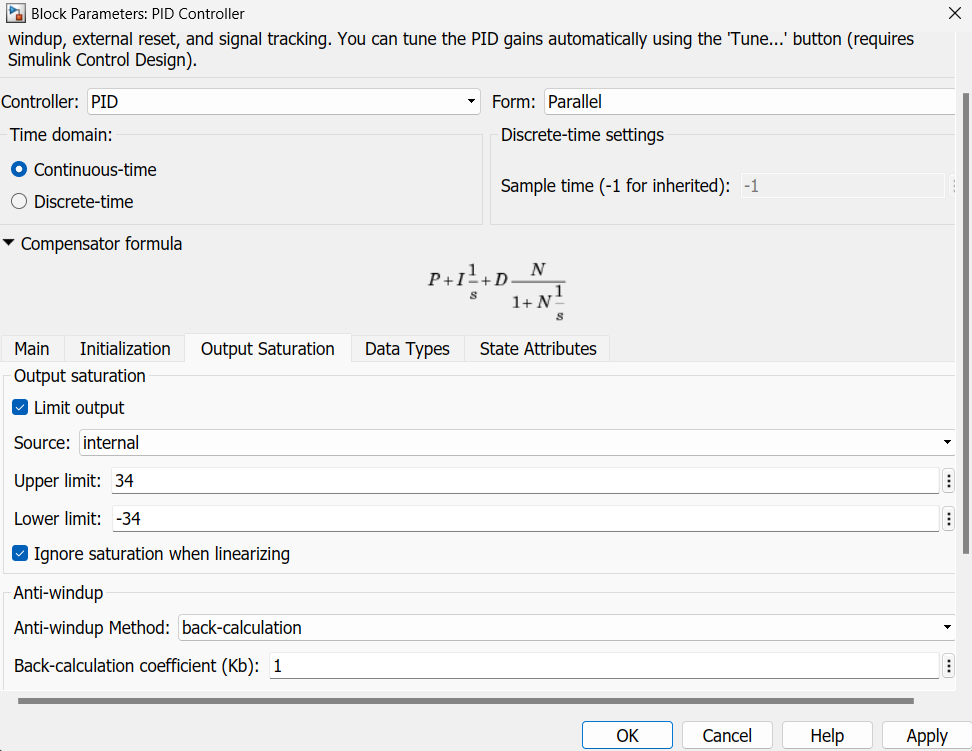


روش انتخاب شده برای Automated Tuning با استفاده از تابع تبدیل انجام می شود که در واقع G(s) گذاشته شده در ادامه و فیدبک تاثیر خود را بر روی ست کردن پارامتر و مقدار های آن ها دارند.

تا به اینجای کار کنترلر PID و تابع تبدیل G(s) که دربرنامه اول داشتیم شبیه سازی شد. همچنین پارامتر های PID نیز مانند برنامه اول ست شد که مقادیری بسیار نزدیک به برنامه اول را دارند.

حال در ادامه می خواهیم پیاده سازی روش آنتی وینداپ را در Simulink توضیح دهیم.

برای این منظور در تنظیمات PID Controller به بخش Output Saturation می رویم:

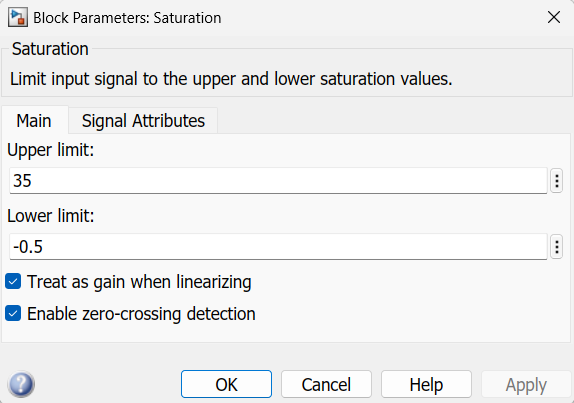
**

*گزینه limit output را تیک زده و برای سادگی بر روی internal نگه می داریم و مقدار آن را با توجه به حدودا نصف اندازه نشان داده شده در خروجی ست می کنیم و بخش مهم ست کردن آنتی وینداپ هست که باید با روش Back Calculation انجام شود*

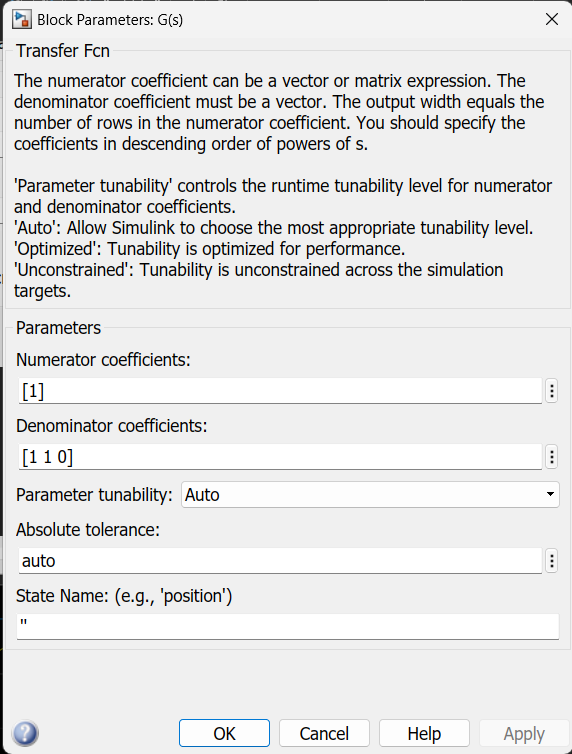
*این ضریب Kb که در اینجا دیده می شود همان 1 / Ta هست و عملا داریم: Kb = 1/Ta*

*مقادیر مناسب برای Ta از 1 تا هر مقدار مثبتی که بخواهیم هست که همین نتیجه می دهد که Kb باید مقداری بین 0 و 1 داده شود.*

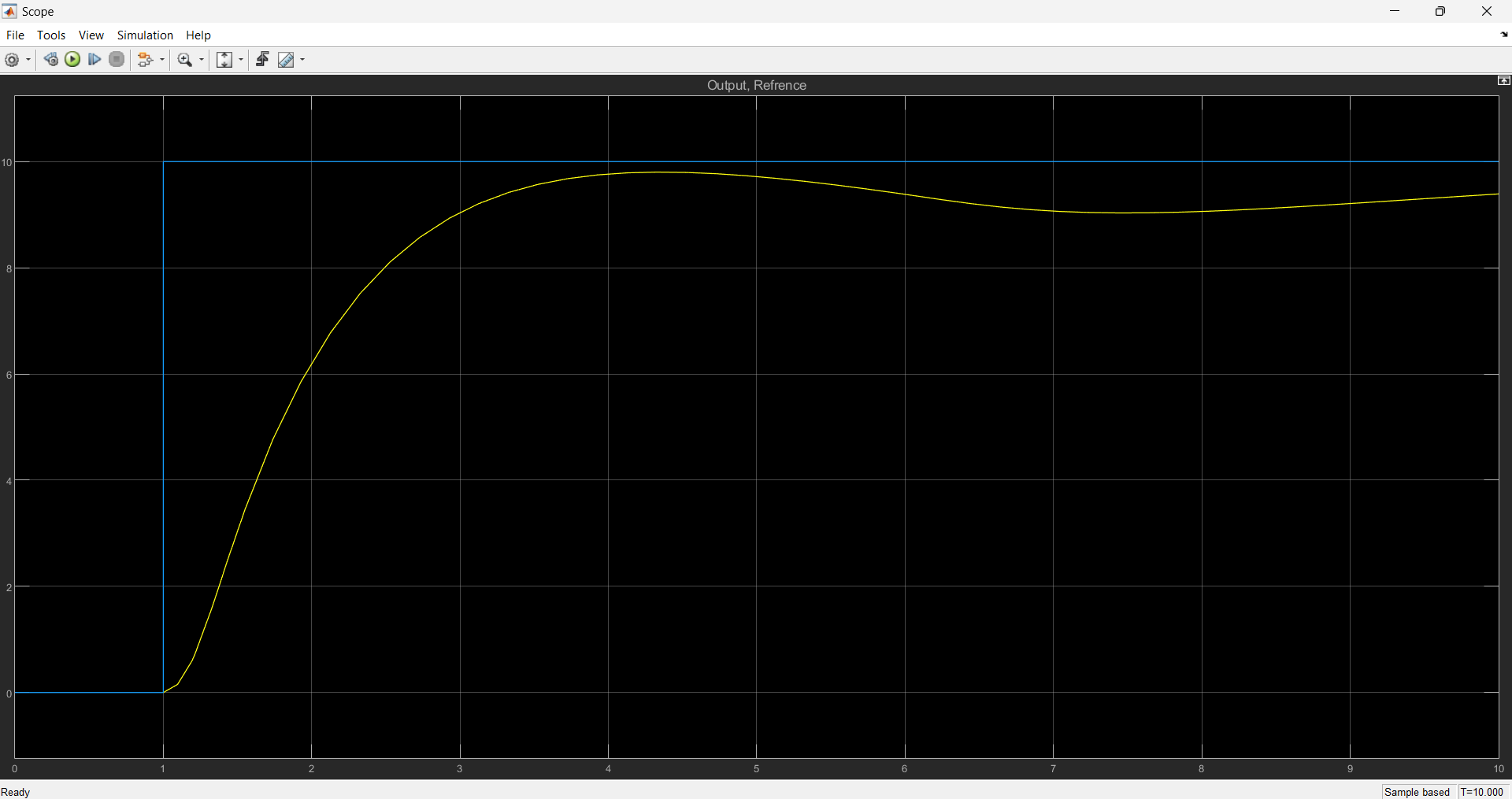
*برای Saturation هم مقدار زیر را ست می کنیم:*

**

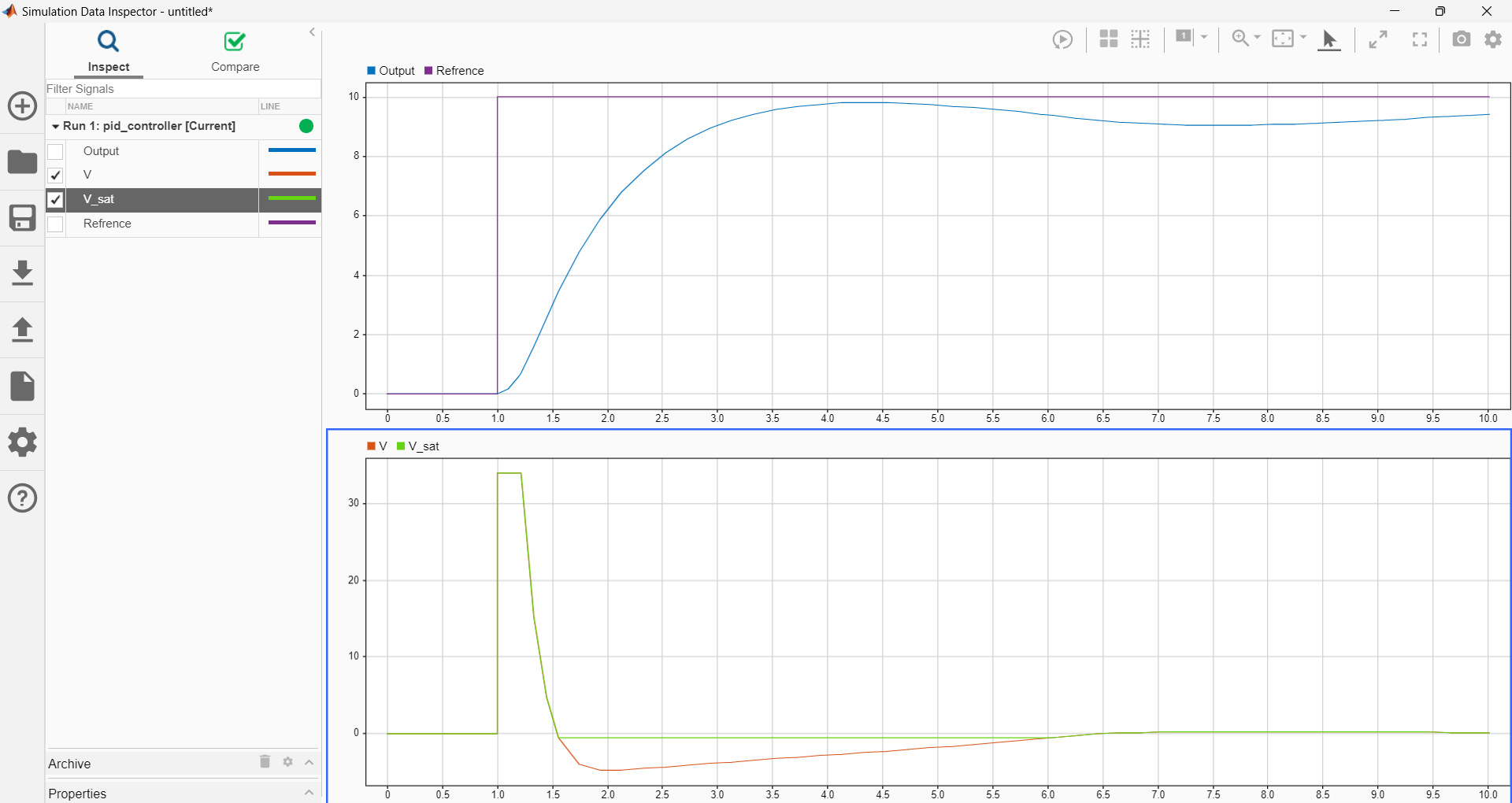
*برای ساخت تابع G(s) هم باید مقادیر زیر را قرار دهیم:*

**

*در نهایت با استفاده از اسکوپ خروجی و ورودی پله را بر روی اسکوپ می توان دید:*

**

*همچنین با توجه به نام گذاری و متغییر دادن به هر محل اتصال این نمودار و نمودار مقایسه V و V\_sat می توان آن را در Inspector نمایش داد:*

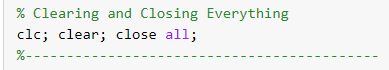
**

*بخش Simulink نشان داده شده فقط برای پیاده سازی آنتی وینداپ به صورت بلوک دیاگرام بوده و خروجی آن را می توان بررسی کرد اما برای نشان دادن و دیدن تغییرات Ta و تاثیرات آن بر روی ناحیه اشباع برنامه دوم در script ای به نام k2.m نوشته شده است.*

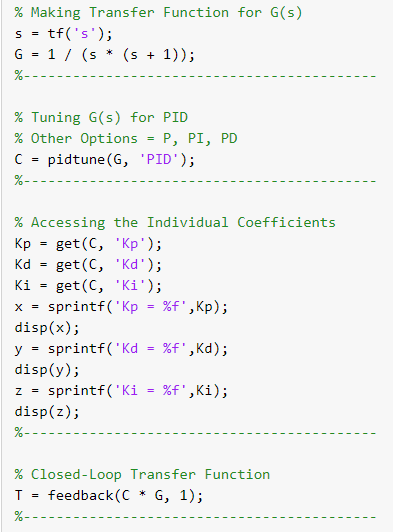
*در ادامه به توضیح برنامه دوم و توضیح نمودارهای آن می پردازیم.*

***بخش برنامه دوم:***

*ابتدا تمامی برنامه ها و خروجی های قبلی را می بندیم و پاک می کنیم:*

**

*در بخش بعدی درست مثل برنامه اول که نوشته بودیم تابع G(s) و C را تشکیل می دهیم ضرائب PID هم مانند قبل با تابع pidtune ست می کنیم و پس از نمایش ضرایب* Kp، Kd و Ki، تابع تبدیل T را با فیدبک 1 و C\*G تشکیل می دهیم (درست مانند برنامه اول که توضیحات را دادیم):



*مقادیر* Kp، Kd و Ki به شکل زیر هستند:

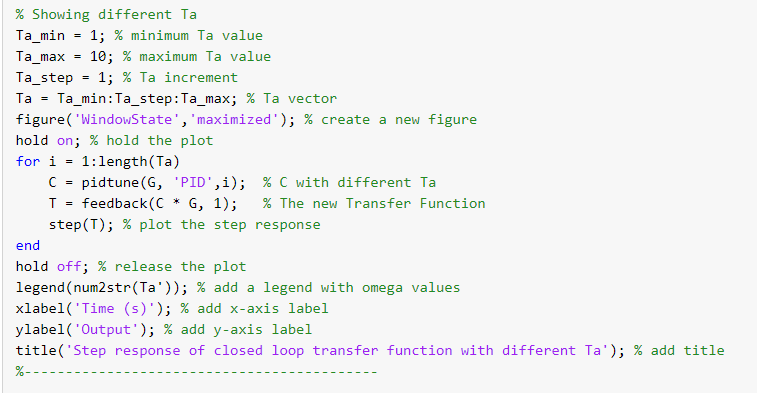
Kp = 2.109126

Kd = 1.834131

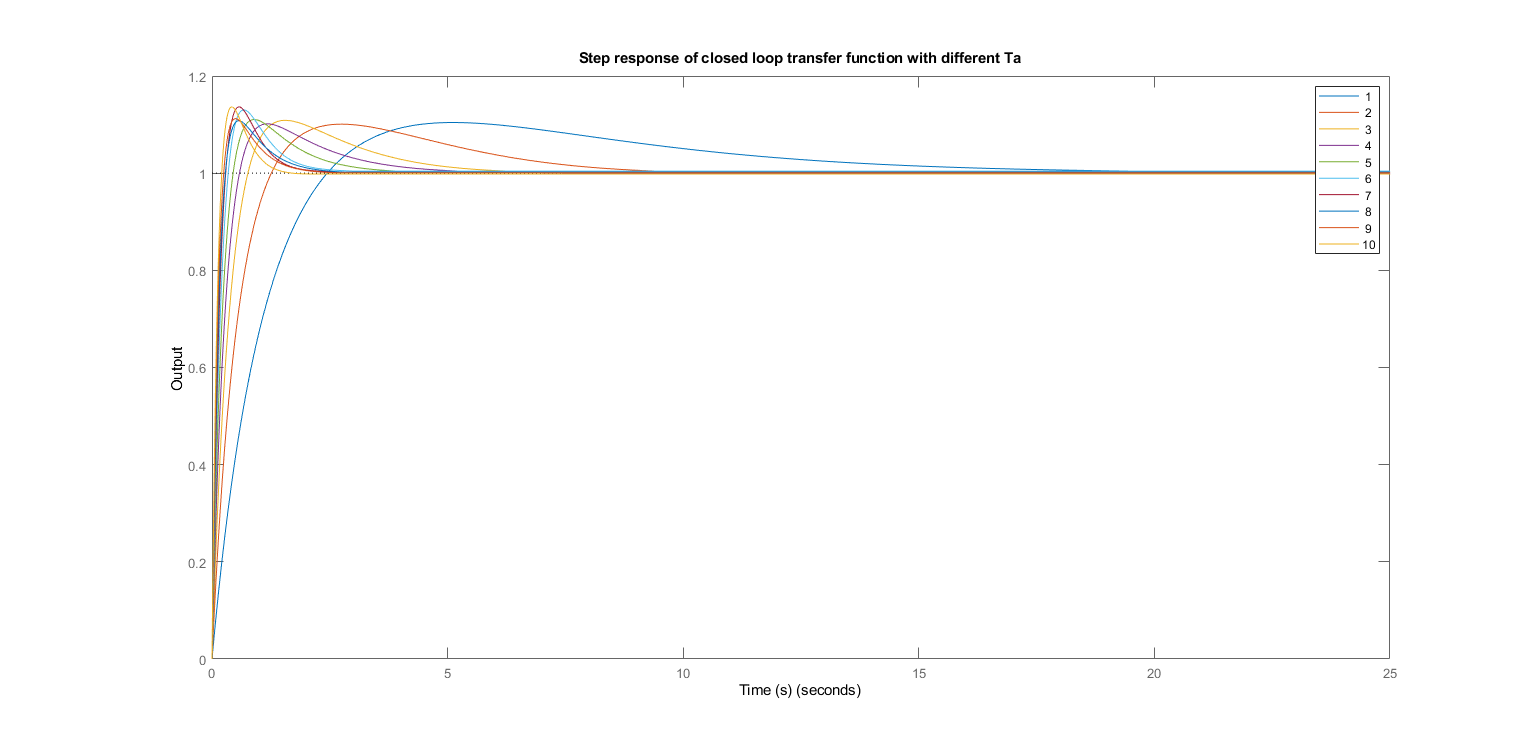
Ki = 0.538966

*در ادامه در بخش آخر برنامه دوم مقادیر مینیمم (1) و ماکسیمم (10) را برای Ta تعیین می کنیم و با استفاده از گام Ta (1) حلقه for را طی می کنیم.*

*در حلقه for پارامتر سوم pidtune که همان Ta هست را برابر با مقدار i در هر دور لوپ قرار می دهیم و تابع C را دوباره با مقدار جدید Ta تیون می کنیم و در نهایت دوباره تابع تبدیل T را دوباره بازسازی می کنیم و نمودار پاسخ ضربه آن T خاص با Ta مورد نظر را رسم می کنیم:*

**

*حال نمودار نهایی را مشاهده می کنیم که به شکل زیر رسم شده است:*

**

در این نمودار می بینیم که هر چه مقدار Ta بزرگتر می شود به عبارتی Kb کوچکتر می شود نمودار رسم شده در ناحیه اشباع به مقدار تابع پله خیلی نزدیک می شود و عملا اثر انتگرال گیری های زیادی کم می شود و تابع رفتاری بسیار نزدیک به پله خواهد داشت.

آن قوس اولیه ای که وجود دارد ناشی از بخش PD کنترل PID هست و عملا می توان گفت که با افزایش Ta رفتار PID را به PD نزدیک می کنیم با این تفاوت که ترم ثابتی که بر اثر خطا در PD وجود دارد عملا در PID جبران می شود و رفتار PID با وینداپ در کاربرد های مختلف برای حذف نیروهای ثابت مثل گرانش و disturbance های مثل باد، زلزله و شوک و .... بسیار کاربردی هست و عملا با وجود سیستم آنتی وینداپ می توان تا حد خوبی از اشباع شدن انتگرال گیر در سیستم PID جلوگیری کرد.