

## سوال ۱

شبیه سازی یک مثال (Real-Time Scheduling) در True Time در نرم افزار متلب و ارایه یک گزارش (رجوع به اسلاید شماره دو)

## جواب سوال ۱

## گزارش کار با نرم افزار MATLAB و اجرای TrueTime

در ابتدا، به دایرکتوری مورد نظر برای فعال سازی TrueTime در متلب مراجعه کردیم با این دستور:

```
>> cd('C:\Users\Iman\Downloads\Reposetories\CE462-ES\Assignments\Assignment 6\truetime-2.0')
>> |
```

سپس، ماژول TrueTime با استفاده از دستور زیر فعال شده است:

```
>> init_truetime
```

برای مشاهده محتوای دایرکتوری فعلی، از دستور ls استفاده شده است:

```
>> ls
.          COPYING      README      examples   kernel
..         HISTORY      docs       init_truetime.m
```

بعد از آن، به دایرکتوری examples و سپس به زیر دایرکتوری threeservos مراجعه شده است. در این قسمت، با کلیک روی فایل با فرمت slx، شبیه سازی threeservos اجرا شده است.

```
>> cd examples\
>> cd threeservos\
|
```

## بررسی فایل‌های داخل فولدر threeservos

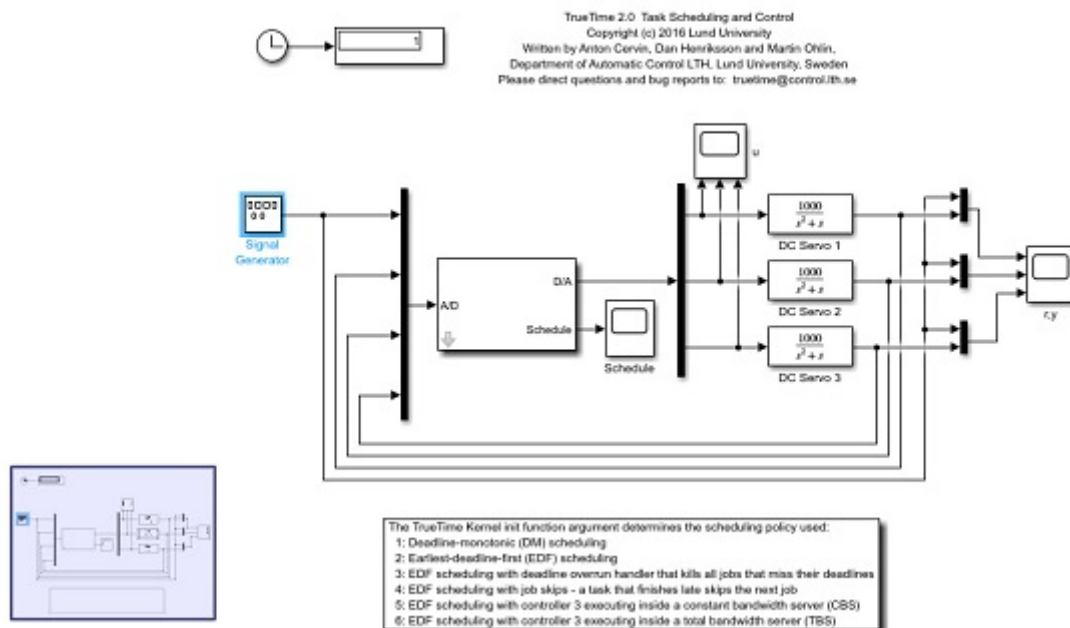
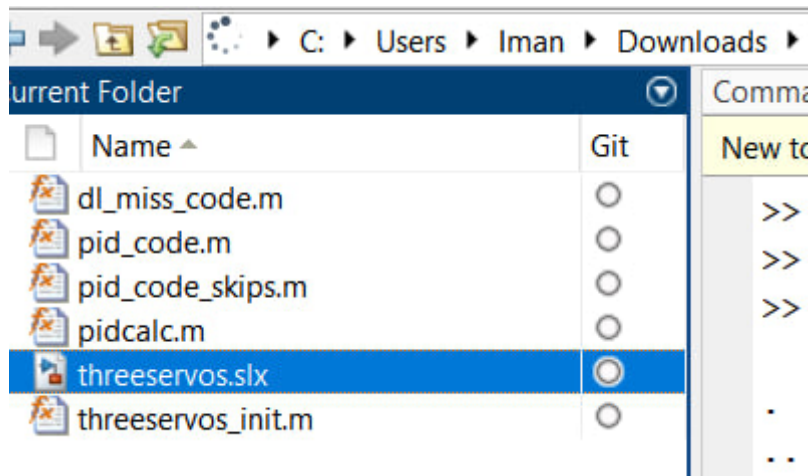
با توجه به فایل‌های موجود در فولدر و فرمت‌ها و نام‌های موجود، فایل threeservos.slx فایل اصلی برنامه است. فایل‌های با پسوند .slx فایل‌های مدل‌سازی Simulink در MATLAB هستند و معمولاً برای شبیه‌سازی سیستم‌های کنترلی و دینامیکی استفاده می‌شوند.

فایل‌های دیگر با پسوند .m همگی اسکریپت‌ها یا توابع MATLAB هستند که توسط فایل threeservos.slx فراخوانی می‌شوند یا به تنهایی اجرا می‌شوند.

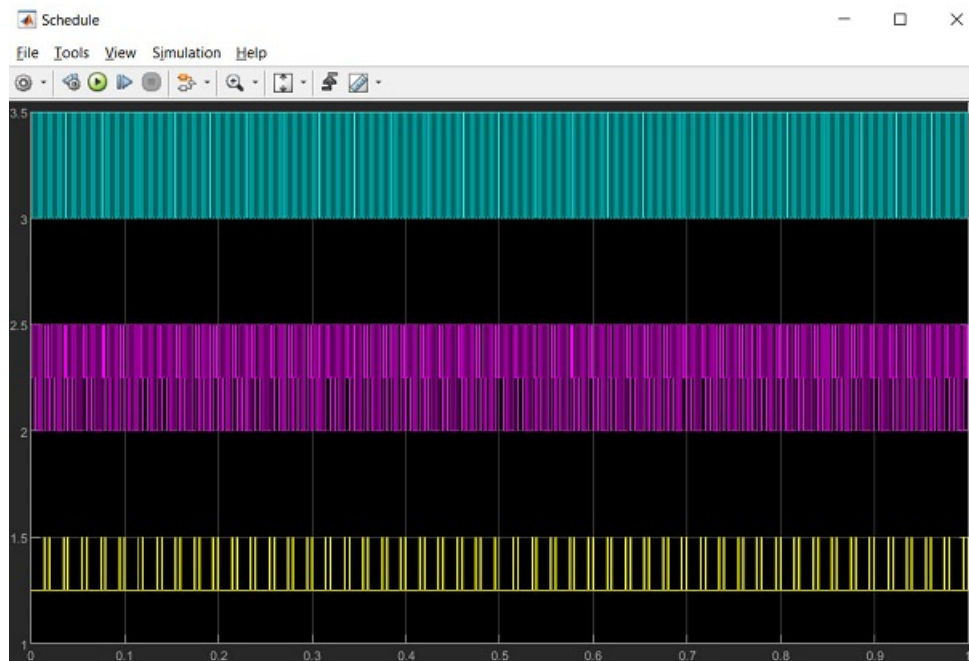
Name	Date modified	Type	Size
 dl_miss_code.m	۱۳۹۵/۰۱/۱۷ ظ.ب. ۱۱:۴۷	MATLAB Code	1 KB
 pid_code.m	۱۳۹۵/۰۱/۱۷ ظ.ب. ۱۱:۴۷	MATLAB Code	1 KB
 pid_code_skips.m	۱۳۹۵/۰۱/۱۷ ظ.ب. ۱۱:۴۷	MATLAB Code	1 KB
 pidcalc.m	۱۳۹۵/۰۱/۱۷ ظ.ب. ۱۱:۴۷	MATLAB Code	1 KB
 threeservos.slx	۱۳۹۵/۰۱/۱۸ ق.ظ. ۱۱:۳۰	Simulink Model	11 KB
 threeservos_init.m	۱۳۹۵/۰۱/۱۷ ظ.ب. ۱۱:۴۷	MATLAB Code	2 KB

## محتویات فایل threeservos.slx

برای تشخیص دقیق‌تر، فایل threeservos.slx را در MATLAB باز کردیم و به محتوای آن نگاهی می‌اندازیم.



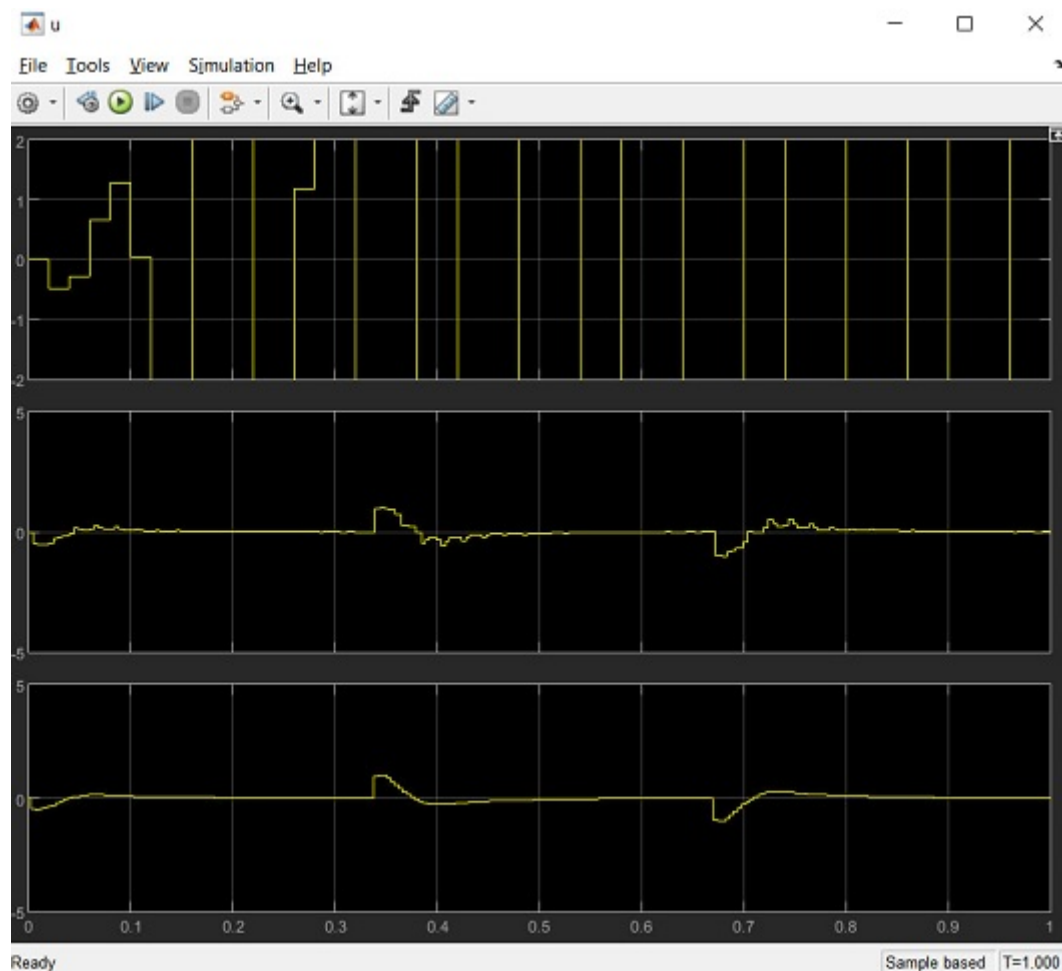
## تصاویر ران برنامه



The Simulink interface shows the 'threeservos' model. The 'Block Parameters: TrueTime Kernel' dialog box is open, displaying the following parameters:

- Subsystem (mask) (link)
- Parameters
- Name of init function (MEX or MATLAB): `threeservos_init`
- Init function argument (arbitrary struct): `1`
- Number of analog inputs and outputs: `[4 3]`
- Number of external triggers: `0`
- (Network and) Node number(s): `[]`
- Local clock offset and drift: `[0 0]`
- ☒ Show Schedule output port
- ☐ Show Energy supply input port
- ☐ Show Power consumption output port

The background shows the Simulink model with a 'TrueTime Kernel' block connected to an 'A/D' block. The status bar at the bottom indicates 'Ready', '125%', and 'ode45'.



Block Parameters: DC Servo 3

**Transfer Fcn**

The numerator coefficient can be a vector or matrix expression. The denominator coefficient must be a vector. The output width equals the number of rows in the numerator coefficient. You should specify the coefficients in descending order of powers of s.

'Parameter tunability' controls the runtime tunability level for numerator and denominator coefficients.  
'Auto': Allow Simulink to choose the most appropriate tunability level.  
'Optimized': Tunability is optimized for performance.  
'Unconstrained': Tunability is unconstrained across the simulation targets.

Parameters

Numerator coefficients:  
[1000]

Denominator coefficients:  
[1 1 0]

Parameter tunability: Auto

Absolute tolerance:  
auto

State Name: (e.g., 'position')  
"

OK Cancel Help Apply

Block Parameters: DC Servo 2

**Transfer Fcn**

The numerator coefficient can be a vector or matrix expression. The denominator coefficient must be a vector. The output width equals the number of rows in the numerator coefficient. You should specify the coefficients in descending order of powers of s.

'Parameter tunability' controls the runtime tunability level for numerator and denominator coefficients.  
'Auto': Allow Simulink to choose the most appropriate tunability level.  
'Optimized': Tunability is optimized for performance.  
'Unconstrained': Tunability is unconstrained across the simulation targets.

Parameters

Numerator coefficients:  
[1000]

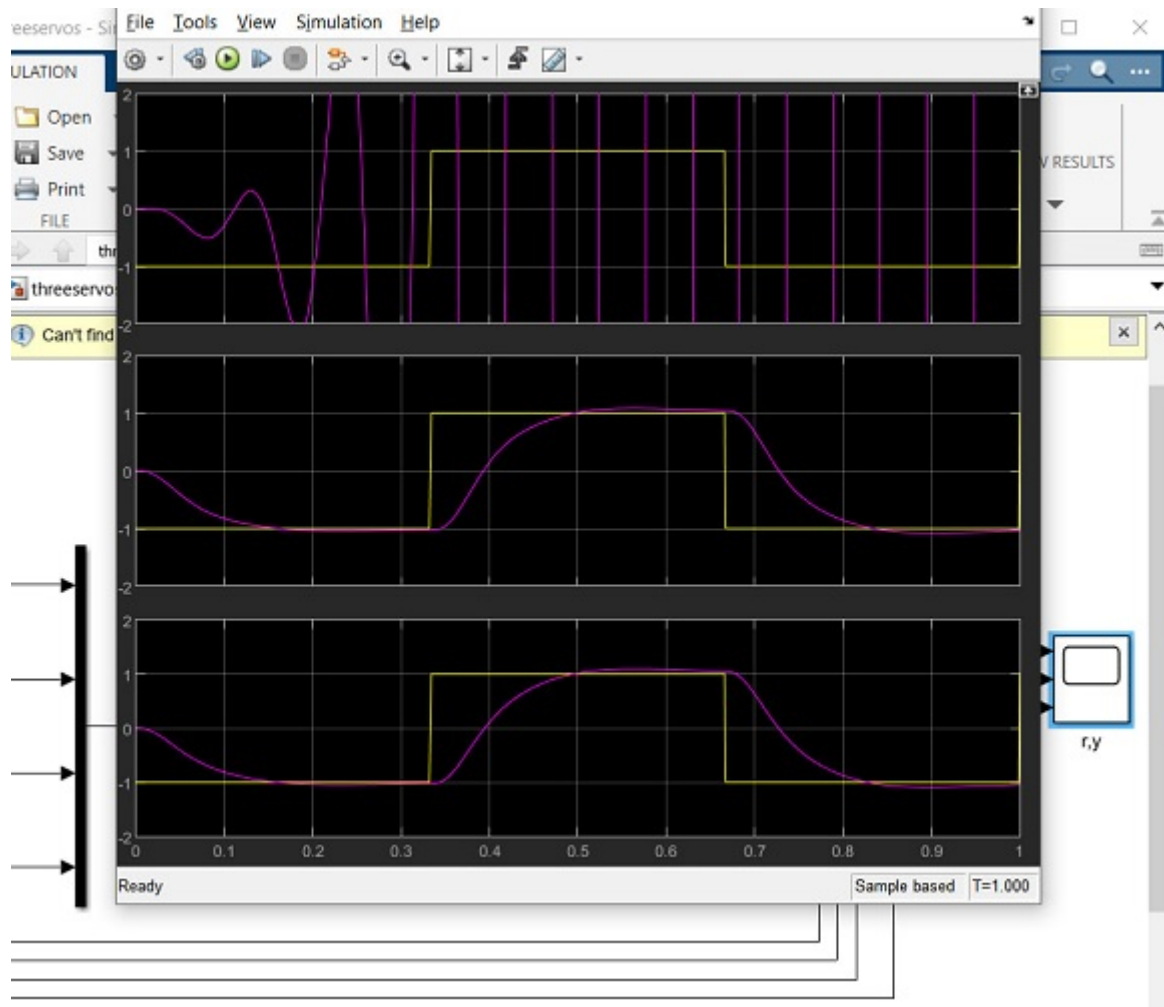
Denominator coefficients:  
[1 1 0]

Parameter tunability: Auto

Absolute tolerance:  
auto

State Name: (e.g., 'position')  
"

OK Cancel Help Apply



## کد threeservos\_init.m

تابع فراخواننده‌ی این شبیه‌سازی در برنامه‌ی متلب:

```
threeservos_init.m
1 function threeservos_init(arg)
2
3 % Task scheduling and control.
4 %
5 % This example extends the simple PID control example (located in
6 % $DIR/examples/servo) to the case of three PID-tasks running
7 % concurrently on the same CPU controlling three different servo
8 % systems. The effect of the scheduling policy on the global control
9 % performance is demonstrated.
10
11 % Initialize TrueTime kernel
12
13 switch arg
14 case 1 % DM scheduling
15     ttInitKernel('prioDM')
16     codefcn = 'pid_code';
17 case {2, % plain EDF scheduling
18       3, % EDF scheduling, kill jobs that overrun
19       5, % EDF scheduling with pid_task3 inside CBS
20       6} % EDF scheduling with pid_task3 inside TBS
21     ttInitKernel('prioEDF')
22     codefcn = 'pid_code';
23 case 4 % EDF scheduling, skip next job if current one late
24     ttInitKernel('prioEDF')
```



## روند ران کردن برنامه

فولدر truetype2 را دریافت می‌کنیم و با اجرای دستور `init_truetype` و سپس وارد این فولدر شدن با دستور `init_motos` و سپس با دستور `soccer` برنامه ران می‌شود.

## توضیح کد

- الف) تعریف تابع: تابعی با نام `soccer_callback` با دو ورودی، `cb` و `N`، تعریف شده است.
- ب) تعریف متغیرها: دو آرایه از رشته‌ها تعریف شده‌اند؛ یکی برای بلوک‌های `Mux` و دیگری برای بلوک‌های `From`.
- ج) حلقه اول: این حلقه برای هر یک از بلوک‌های `Mux` اجرا می‌شود:
- تعداد ورودی‌های فعلی بلوک `Mux` بازیابی می‌شود.
  - یک حلقه داخلی برای هر ورودی بلوک `Mux` اجرا می‌شود و:
    - اتصالات موجود بین بلوک‌های `From` و `Mux` حذف می‌شوند.
    - بلوک‌های `From` موجود حذف می‌شوند.
  - تعداد ورودی‌های بلوک `Mux` به مقدار جدید `N` تنظیم می‌شود.
- د) حلقه دوم: این حلقه برای ایجاد بلوک‌های جدید از نوع `From` و اتصال آن‌ها به بلوک‌های `Mux` اجرا می‌شود.
- برنامه به اتمام می‌رسد پس از اجرای دو حلقه‌ی `for` که در آنها، تمامی بلوک‌های مرتبط حذف شده و بلوک‌های جدید اضافه می‌شوند. پس از اجرای کامل دومین حلقه، تابع به پایان خواهد رسید.



## آموزش نصب و فعال‌سازی MATLAB ورژن 2023a

الف) قبل از شروع نصب، اتصال به اینترنت را قطع می‌کنیم.

ب) فایل مورد نظر از حالت فشرده خارج شده است.

ج) فایل R2023a\_Windows.iso را با استفاده از یک برنامه‌ی درایو مجازی Mount نموده و نصب را شروع می‌کنیم.

د) فایل Setup را اجرا کرده و در قسمت Enter File Installation Key، سریال گفته شده را وارد کردیم:

ه) در مرحله Select License File، فایل license.lic واقع در پوشه‌ی Crack را انتخاب می‌کنیم.

و) پس از نصب، نرم‌افزار را اجرا نمی‌کنیم.

ز) فایل libmwlmgrimpl.dll را از پوشه‌ی Crack به مسیر گفته شده کپی کردیم و فایل را در آن مسیر جایگزین می‌نماییم. کپی کرده و فایل موجود در آن مسیر را جایگزین می‌کنیم.

ح) حال می‌توانید نرم‌افزار را اجرا کنیم.

ط) در صورت نیاز به آپدیت، فایل آپدیت با فرمت iso را مانت کرده و فایل Update.cmd را اجرا می‌کنیم.

ی) مجدداً فایل libmwlmgrimpl.dll را جایگزین می‌کنیم.