به نام خدا

تمرین شماره 1

درس کنترل سیستم های عصبی عضلانی

تهیه کننده: علیرضا امیری

شماره دانشجویی: 40202414

استاد درس: دکتر دلربایی

پاییز 1402

1. کاربرد های عملی MJT در دنیای واقعی

**فعالیت های روزمره:**

1. *نوشتن*: در فرایند نوشتن، دست انسان با استفاده از MJT، مسیری هموار و بدون تنش را طی می کند و در نتیجه، خطوط رسم شده بر روی کاغذ بسیار طبیعی به نظر می رسند.
2. *راه رفتن*: به هنگام راه رفتن، پا به گونه ای حرکت می‌کند و تاب می‌خورد که مسیر طی شده دارای کمترین جرک باشد.
3. *غذا خوردن*: برای برداشتن غذا از روی میز و رساندن قاشق به دهان، معمولا دست انسان مسیری با کمینه جرک طی می‌کند.
4. ساز زدن: در هنگام ساز زدن، معمولا انگشتان نوازنده برای جابه‌جایی بین موقعیت های مختلف، مسیری با کمینه جرک را طی می‌کند.

کاربردهای عملی MJT:

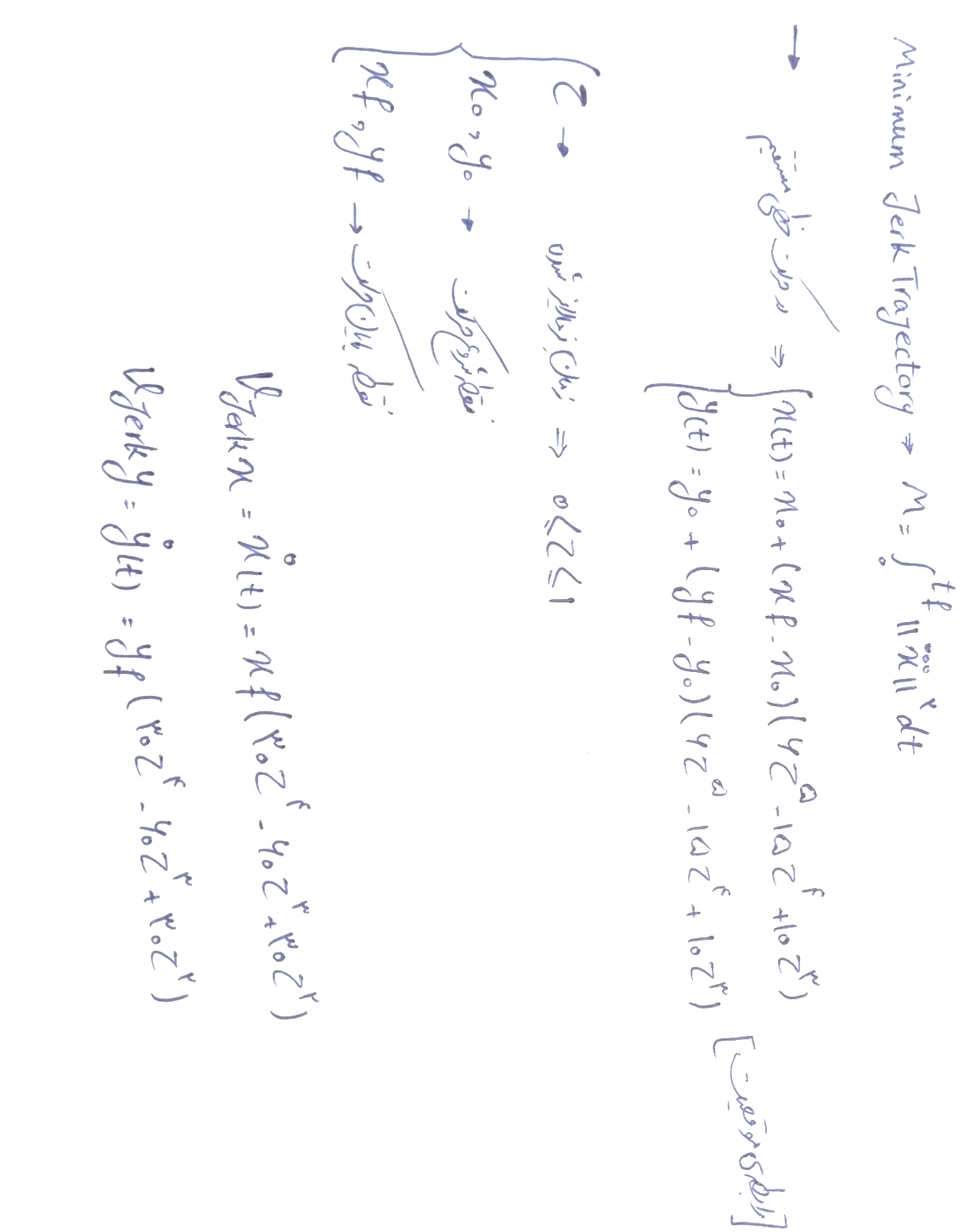
1. بازو های رباتیک: در بازو های رباتیک، قسمت ابزار بازو باید مسیر بین نقاط مختلف را طی کند. بنابراین، با اعمال MJT، می‌توان مسیری بازو را طوری طراحی کرد که برای جابه‌جایی بین نقاط مختلف، مسیری بهینه و هموار را طی کند.
2. سیستم های همکاری انسان و ربات (human-robot cooperation):

در این سیستم ها، از آنجایی که بازوی رباتی باید بتواند کاملا با حرکت دست انسان هماهنگ باشد، باید قادر باشد مانند دست انسان، مسیری هموار و بدون تنش را طی کند. بنابراین، در این سیستم ها از MJT استفاده می‌شود.

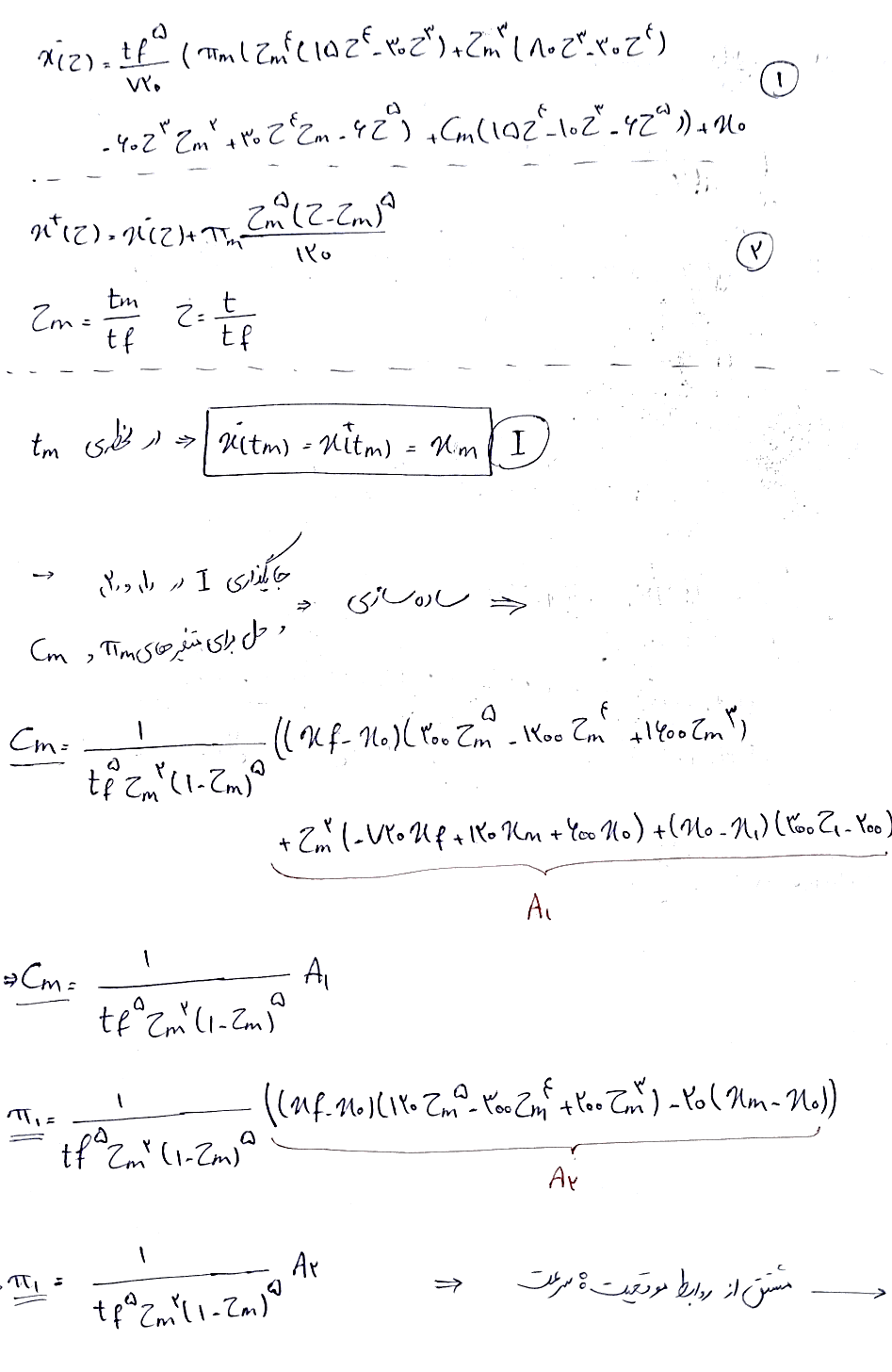
1. کاربرد های لمسی (Haptic Application):

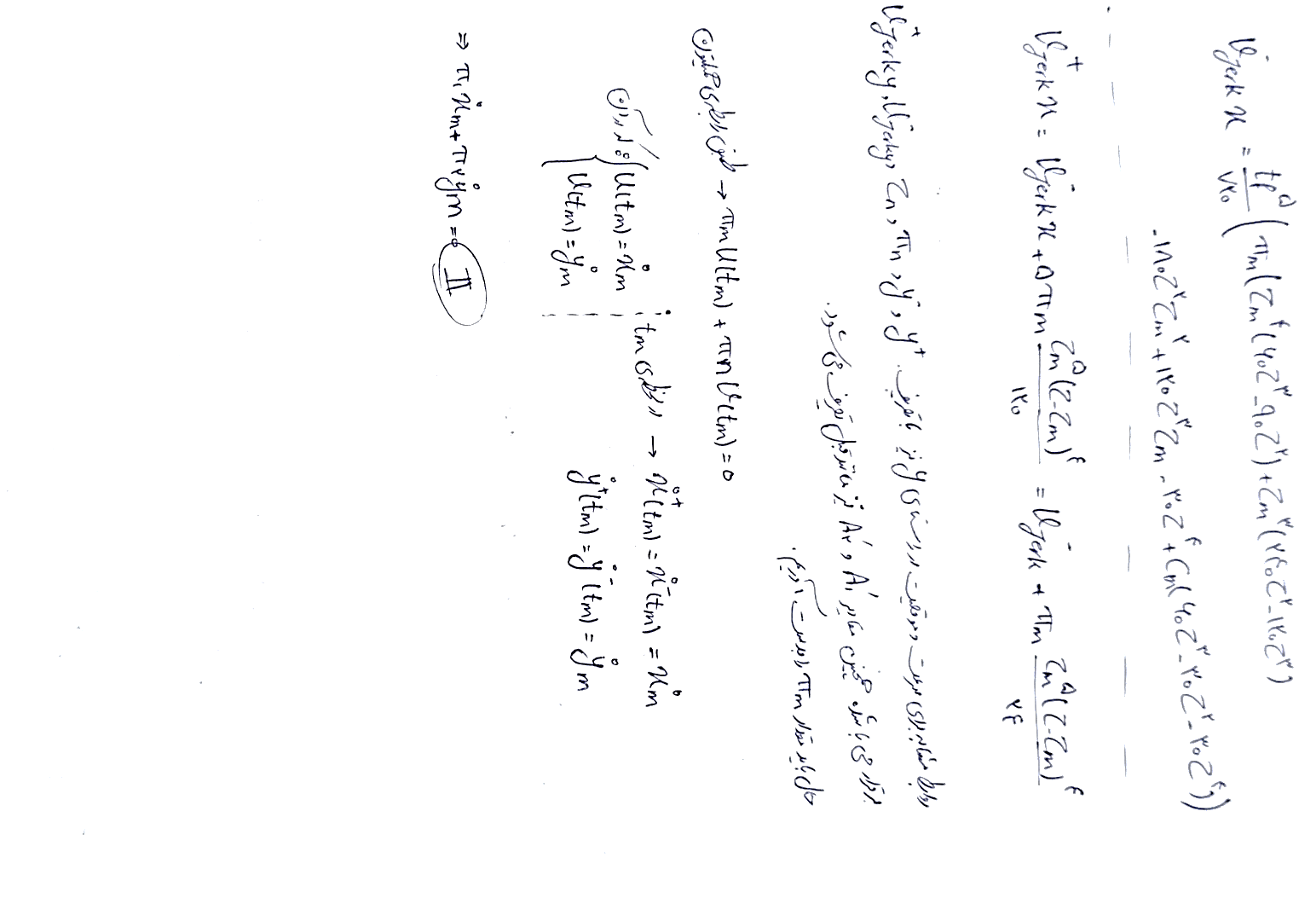
در فرایند توان بخشی، برای بیمارانی که بر اثر سکته توانایی کنترل بخشی از ماهیچه های دست و پای خود را از دست داده‌اند، از MJT به منظور طراحی حرکت های مانند انسان (human-like) بهره برده می‌شود.

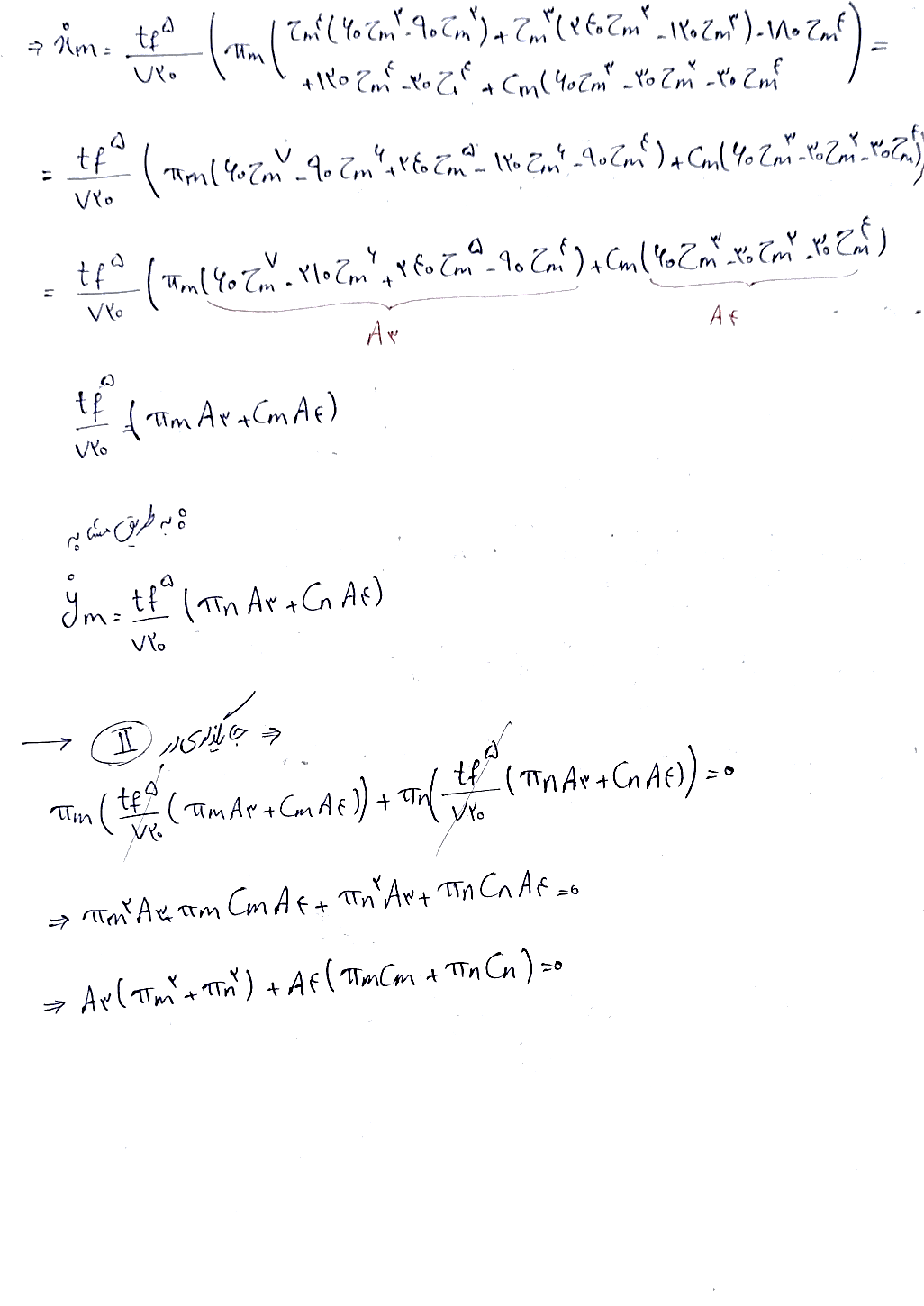
1. رابطه‌ی موقعیت و سرعت و MJT در حرکت خطی مستقیم

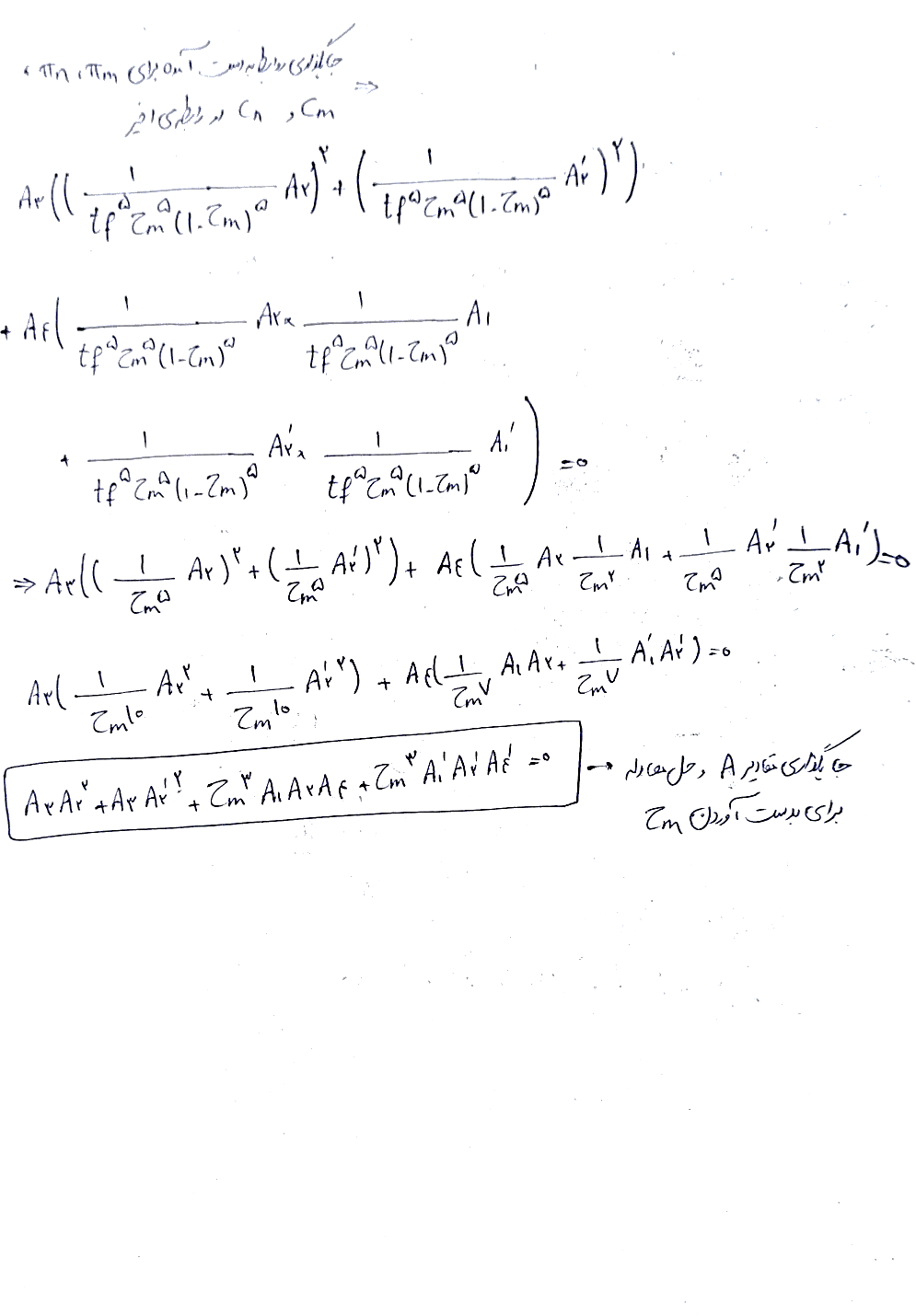


1. رابطه‌ی موقعیت و سرعت و MMJT در حرکت منحنی‌الخط









کد متلب برای حل معادله ی فوق:

clc

clear all;

syms t1 tf xf yf x1 y1 float

t1 = sym('t1', 'real');

tf = sym('tf', 'real');

xf = sym('xf', 'real');

yf = sym('yf', 'real');

x1 = sym('x1', 'real');

y1 = sym('y1', 'real');

xf = 0.6

yf = 0.0

x1 = 0.1

y1 = 0.3

%%

A2 = expand(xf\*(120\*t1^5 - 300\*t1^4 + 200\*t1^3) -20\*x1)

A22 = expand(yf\*(120\*t1^5 - 300\*t1^4 + 200\*t1^3) -20\*y1)

A1 = expand(xf\*(300\*t1^5 - 1200\*t1^4 + 1600\*t1^3) + t1^2\*(-720\*xf + 120\*x1) - x1\*(300\*t1 - 200))

A11 = expand(yf\*(300\*t1^5 - 1200\*t1^4 + 1600\*t1^3) + t1^2\*(-720\*yf + 120\*y1) - y1\*(300\*t1 - 200))

A3 = 60\*t1^7 - 210\*t1^6 + 240\*t1^5 - 90\*t1^4

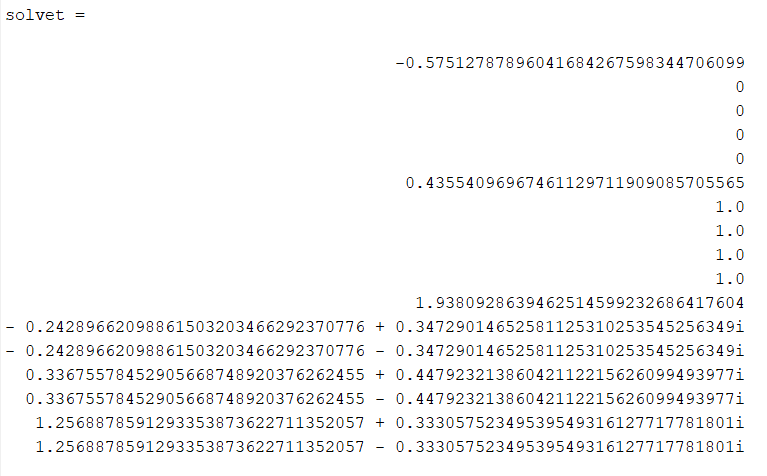
A4 = 60\*t1^3 - 30\*t1^2 - 30\*t1^4

Result = expand(A3\*A2\*A2 + A3\*A22\*A22 + (t1^3)\*A1\*A2\*A4 + (t1^3)\*A11\*A22\*A4)

%numerical solution

solvet = vpasolve(Result, t1)

پس از حل معادله به وسیله‌ی متلب و پیدا شدن ریشه های معادله، تنها ریشه‌ای را می‌پذیریم که مقدار آن نرمالایز باشد، یعنی عددی میان 0 و 1.



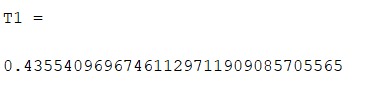
در این مثال، با در نظر گرفتن مقادیر اولیه‌‌ی مفروض، تنها مقدار 0.435 قابل قبول می‌باشد.

برای انتخاب عدد حقیقی که میان دو مقدار 0 و 1 باشد، به وسیله ی دو دستور زیر ریشه ها را فیلتر می‌کنیم.

real\_roots = solvet(imag(solvet)==0); % filter out only real roots

T1 = real\_roots(real\_roots > 0 & real\_roots < 1) % filter out roots between 0 and 1

در نهایت، ریشه‌ی مورد نظر در متغیر T1 ذخیره می‌شود.



* رسم نمودار های موقعیت و سرعت در راستا های X و Y

با مشخص شدن زمان رسیدن به نقطه‌‌ی میانی، می‌توانیم از تابع های موقعیت و سرعت جسم استفاده کرده و با جایگذاری مقادیر T1، در هر زمان t موقعیت و سرعت آن را به دست آوریم. رابطه های مورد نیاز در این قسمت، در محاسبات بالا به با مستطیل قرمز مشخص شده اند.

در گام اول، این روابط در محیط متلب نوشته می‌شوند.

1. روابط مربوط به موقعیت:
2. %% X-Position Equations as a function of time
3. pi1 = 1/(tf^5 \* t1^5 \* ((1-t1)^5)) \* A2;
4. c1 = 1/(tf^5 \* t1^2 \* ((1-t1)^5)) \* A1;
5. star1(t) = t1^4 \* (15\*t^4 - 30\*t^3)+t1^3 \* (80\*t^3-30\*t^4)-60\*t^3\*t1^2+30\*t^4\*t1-6\*t^5;
6. star2(t) = 15\*t^4 - 10\*t^3 - 6\*t^5;
7. x\_negative(t,t1) = (tf^5)/720 \* ((pi1\*star1)+ c1 \*star2);
8. x\_positive(t,t1) =x\_negative(t,t1) + pi1\* (t1^5 \* ((t-t1)^5))/120;
9. Position\_X(t,t1) = heaviside(t1 - t) \* x\_negative(t,t1) +heaviside(t - t1) \* x\_positive(t,t1) ;
10. %% Y-Position Equations as a function of time
11. pi2 = 1/(tf^5 \* t1^5 \* ((1-t1)^5)) \* A22;
12. c2 = 1/(tf^5 \* t1^2 \* ((1-t1)^5)) \* A11;
13. y\_negative(t,t1) = (tf^5)/720 \* ((pi2\*star1)+ c2 \*star2);
14. y\_positive(t,t1) = y\_negative(t,t1) + pi2\* (t1^5 \* ((t-t1)^5))/120;
15. Position\_Y(t,t1) = heaviside(t1 - t) \* y\_negative(t,t1) +heaviside(t - t1) \* y\_positive(t,t1) ;

2. روابط مربوط به سرعت:

%% X-Velocity Equations as a function of time

Vx\_negative = tf^5/720\*(pi1\*(t1^4\*(60\*t^3-90\*t^2)+t1^3\*(240\*t^2-120\*t^3)-180\*t^2\*t1^2+120\*t^3\*t1-30\*t^4)+c1\*(60\*t^3-30\*t^2-30\*t^4));

Vx\_positive = Vx\_negative + pi1/24\*(t1^5\*(t-t1)^4);

Velocity\_X(t,t1) = heaviside(t1 - t) \* Vx\_negative + heaviside(t - t1) \* Vx\_positive;

%% Y-Velocity Equations as a function of time

Vy\_negative = tf^5/720\*(pi2\*(t1^4\*(60\*t^3-90\*t^2)+t1^3\*(240\*t^2-120\*t^3)-180\*t^2\*t1^2+120\*t^3\*t1-30\*t^4)+c2\*(60\*t^3-30\*t^2-30\*t^4));

Vy\_positive = Vy\_negative + pi2/24\*(t1^5\*(t-t1)^4);

Velocity\_Y(t,t1) = heaviside(t1 - t) \* Vy\_negative + heaviside(t - t1) \* Vy\_positive;

در نهایت، هر یک از نمودار های مورد نظر به وسیله‌ی دستور fplot در بازه‌ی [0,tf] رسم می‌شوند.

1. الف) حرکت در راستای y

X0 = y0 = xf = 0

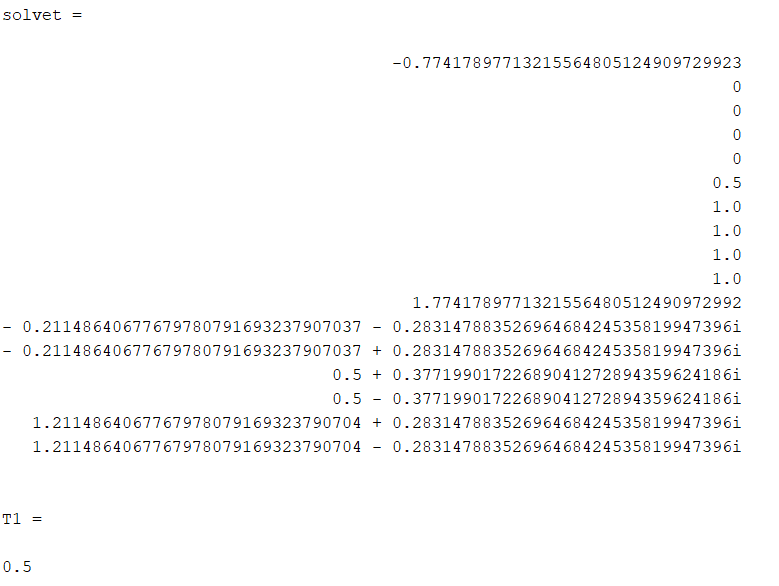
Yf = 0.4m

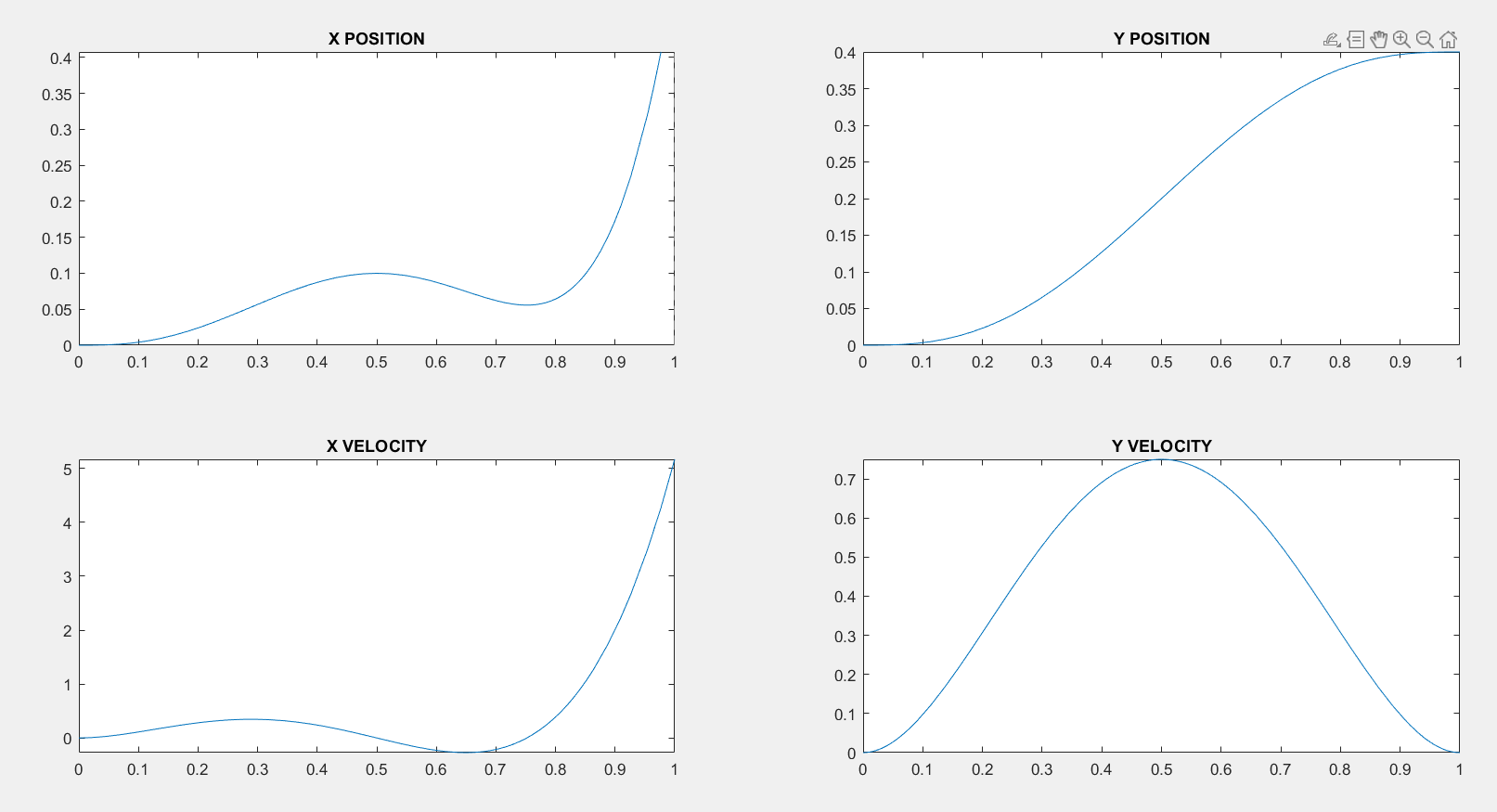
xm = 0.1m

ym = 0.2m

---------------------------------------------------------------------------------------

با جایگذاری مقادیر فوق در کد نوشته شده، نتایج زیر حاصل می‌شود.





1. ب) حرکت در راستای x

X0 = y0 = yf = 0

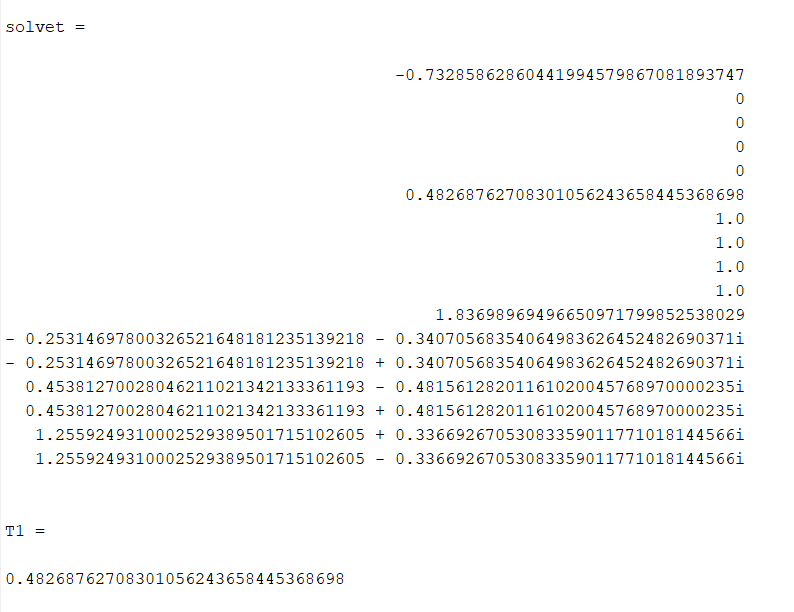
xf = 0.5m

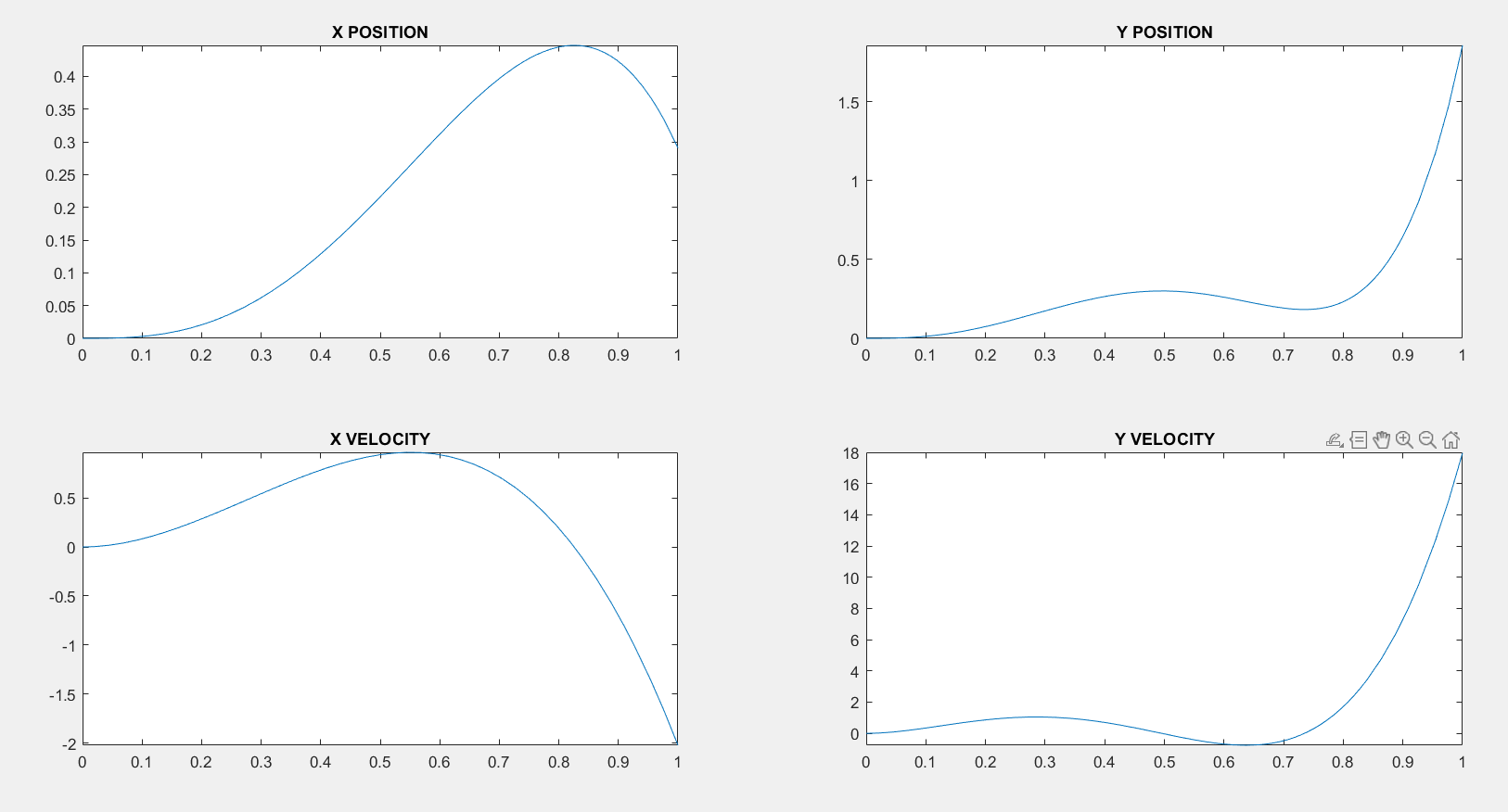
xm = 0.2m

ym = 0.3m

----------------------------------------------------------------------------------------------

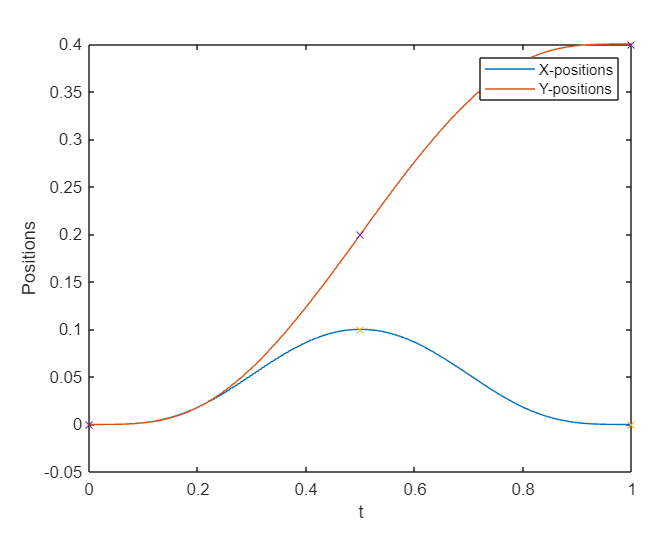
مجدد، با جایگذاری مقادیر فوق در کد، نتایج زیر حاصل می‌شود:

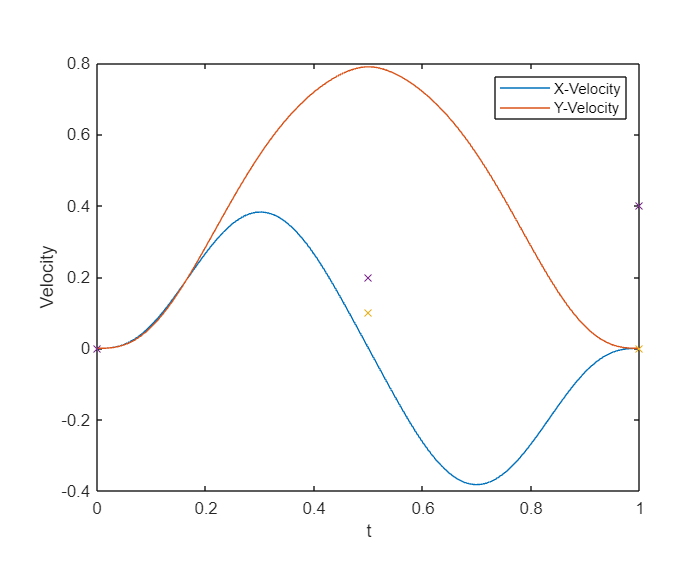


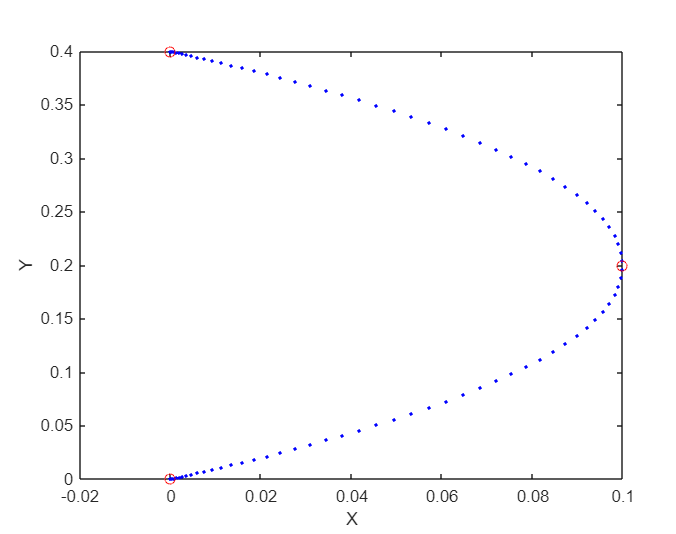


روش دوم حل ( با استفاده از ابزار minjerkpolytraj متلب)

1. حرکت در راستای Y
2. wpts = [0 0.1 0; 0 0.2 0.4];
3. tpts = 0:0.5:1;
4. numsamples = 100;
5. [q,qd,qdd,qddd,pp,timepoints,tsamples] = minjerkpolytraj(wpts,tpts,numsamples);
6. plot(tsamples,q)
7. hold on
8. plot(timepoints,wpts,'x')
9. xlabel('t')
10. ylabel('Positions')
11. legend('X-positions','Y-positions')
12. hold off
13. figure
14. plot(tsamples,qd)
15. hold on
16. plot(timepoints,wpts,'x')
17. xlabel('t')
18. ylabel('Velocity')
19. legend('X-Velocity','Y-Velocity')
20. hold off
21. figure
22. plot(q(1,:),q(2,:),'.b',wpts(1,:),wpts(2,:),'or')
23. xlabel('X')
24. ylabel('Y')







2. حرکت در راستای X

wpts = [0 0.2 0.5; 0 0.3 0];

tpts = 0:0.5:1;

numsamples = 100;

[q,qd,qdd,qddd,pp,timepoints,tsamples] = minjerkpolytraj(wpts,tpts,numsamples);

plot(tsamples,q)

hold on

plot(timepoints,wpts,'x')

xlabel('t')

ylabel('Positions')

legend('X-positions','Y-positions')

hold off

figure

plot(tsamples,qd)

hold on

plot(timepoints,wpts,'x')

xlabel('t')

ylabel('Velocity')

legend('X-Velocity','Y-Velocity')

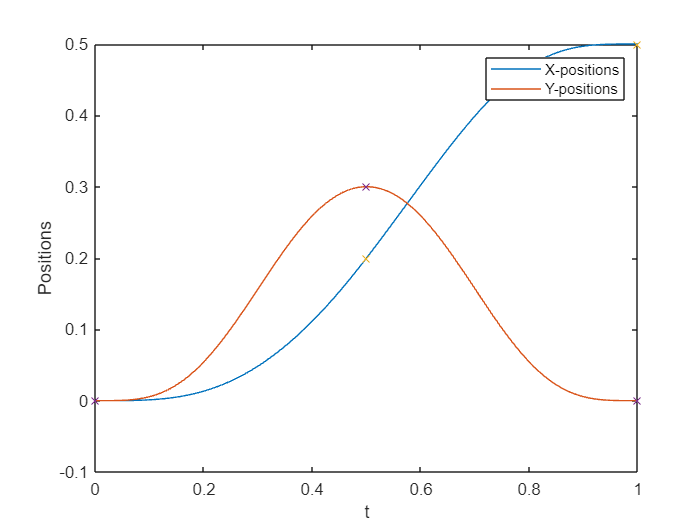
hold off

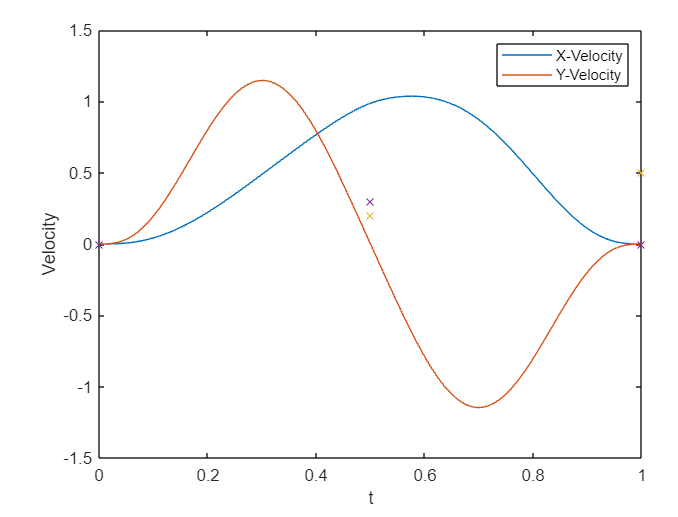
figure

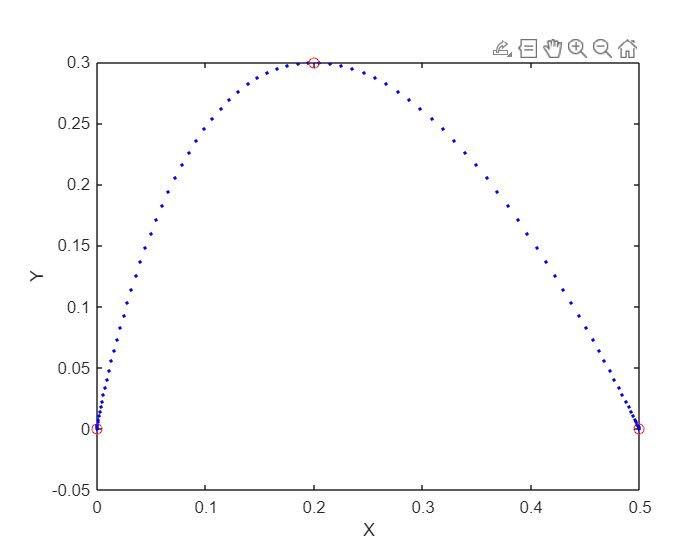
plot(q(1,:),q(2,:),'.b',wpts(1,:),wpts(2,:),'or')

xlabel('X')

ylabel('Y')







به پیوست، فایل های متلب مربوط به این تمرین خدمتتان ارسال می‌شود.   
ممنون از توجه شما