الموضوع   التاريخ: / /

* Robotic systems 6 : Trajectory planning

➤ Trajectory planning in joint space & cartesin space.

q is a pose

$q_i \to q_f$, jTraj

$q_i = [\theta_{1i}; \theta_{2i}; \theta_{3i}; \theta_{4i}; \theta_{5i}; \theta_{6i}]$

$q_f = [\theta_{1f}\ \theta_{2f}\ \theta_{3f}\ \theta_{4f}\ \theta_{5f}\ \theta_{6f}]$

$t = [0:0.01:3]$

لو انا عاوز احرك الربوتة من ال pose الأولى الى التانية من خلال ٣ ثواني على (300) خطوة

اللي هيحصل دلوقتي أنه هيغير

$\theta_{1i} \to \theta_{1f} \xrightarrow{\text{على}} 300$ step

$\theta_{2i} \to \theta_{2f} \longrightarrow 300$ "

↓

$$q_{new} = \begin{bmatrix} \theta_{1i} & \theta_{2i} & \cdots & \theta_{6i} \\ & & & \\ \theta_{1f} & \theta_{2f} & \cdots & \theta_{6f} \end{bmatrix}$$

حتى يصل ال q_{new}

✓ التغير من فوق لتحت

t هيتم في 300 step

✓ خلى بالك هما هيبقو 301 عنصر علشان ال step في الفترة من عشر لأخر

NOTE

لو عملت "inv q" inverse kinematics على (q) وحطيت serial link

$q =$ serial link . ikinematics6s (T)

closed for solution هتدينك

هيطلب T

f kinematics $\xrightarrow{\text{بتحول}}$ q $\xrightarrow{\text{الى}}$ T

I kinematics $\xrightarrow{\text{بتحول}}$ T $\xrightarrow{\text{الى}}$ q

$T =$ serial link . fkine (q) $\longrightarrow$ cartersician space دي تديك ال

SENA

- Robotics Toolbox Example on joint space Trajectory planning

Video on matlab

```
>> mdl_puma560
>> T1 = Transl(0.4, 0.2, 0) * trotx(pi)
>> T2 = Transl(0.4, -0.2, 0) * trotx(pi/2)
>> q1 = p560.ikine6s(T1)
>> q2 = p560.ikine6s(T2)
```

$$T_1 = \begin{matrix} 1 & 0 & 0 & 0.4 \\ 0 & -1 & 0 & 0.2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Homogeneous Transformation بيمثل Pose

$$T_2 = \begin{matrix} 1 & 0 & 0 & 0.4 \\ 0 & 0 & -1 & -0.2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

q_1 = 3.2631 ✓ ✓ ✓ ✓ ✓ جدول زوايا

(2,3) المتغير هيكون في كل الزوايا ما عدا

T_1 لـ joint space الكون ده هيبقى

استعملنا هنا 6s لانه 6 degree of freedom spherical wrist

لو عاوز اشوف شكل الـ pose في حالة q_1 هعمل

```
>> p560.plot(q1)
```

```
>> t = [0 : 0.05 : 2]'
```

(5 msec)

→ هيخليلي الـ Time يكون vector

```
>> length(t)
= 41
```

```
>> q = jtraj(q1, q2, t)
```

→ تنقل في joint space بين q_1 الى q_2 في زمن (t)

> joint space Trajectory: Further work on The Same Example with velocity of acceleration

\# ملحوظة.   Video Matlab

١- عندي حاجة اسمها (sTraj) وحاجة (cTraj)

خاصة ب Joint space ↔ خاصة ب Cartesian space

انا عمل inverse kinematics المركة بيفهم "الوضع الاول والأخير" المعاملة

ب Joint space لأني مش عاوز ادري أنا ازاي بيتحرك ف

cartesian space يهمني الشغل ع $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$

٢- لما جيت عملت زمن شاك $t = [0, 0.05, 2]$

هو مقسم (θ) من ١ لحد 6 على عدد الـ (pose) بالتساوي

لأ... ولكنه بيعمل velocity of acceleration planning

	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
41 demat with 40 step	✓					
	✓					
	↓	↓	↓	↓	↓	↓
	θ_{1f}	θ_{2f}	θ_{3f}	θ_{4f}	θ_{5f}	θ_{6f}

الفرق بين القريب الا محو والي تحت مش متساوي بسبب velocity of acceleration planing

ولكنه اللي لازم اكون فاهمه بيبدأ من zero velocity وينتهي zero velocity

كذلك الوضع للـ acceleration

$q_1 \rightarrow$ row vector $q_2 \rightarrow$ row vector

$q \rightarrow$ 41 row vector

3- $[q, q_d, q_{dd}] = jtraj[q_1, q_2, t]$

خاصة بالـ pose 41*6   خاصة بالسرعة 41*6   خاصة بالعجلة 41*6

٤- فيه command موجود في Toolbox اسمه → qplot(q)

بيبين كيف لتغير الـ $\theta_1 \rightarrow 6$

" qplot(qd) → هيدينا رسمة لل $q_1 \rightarrow q_6$ فيها شكل التغير بتاع السرعة وكذلك لو عملنا (qdd)

وهي دي حلاوة وجمال ال path planning بيدينا نظرة للحركة والتغيرات وبالتالي لو فيه مشكلة اقدر احلها

➤ The steps Required for joint space Trajectory planning

ملخص الخطوات ← Trajectory planning in joint space

1- specify starting pose T_1 4*4
2- " final " T_2 4*4
3- find inverse kinematics for $T_1 \rightarrow (q_1)$ 1*6
ikine6s (6s)
4- " " " " $T_2 \rightarrow (q_2)$ 1*6
(6s)
5- generate a Time vector t(0:0.05:2) 1*41 or 41*1.
6- generate Trajectory in joint space from
q_1 To q_2 in steps of length(t)
q_j = jtraj(q_1, q_2, length t) 41*6
7- [q_j q_{jd} q_{jdd}] = jtraj(q_1 q_2 t)
θ_i ω_i α_i

➤ plotting The xy path Based on joint space Trajectory planning

يوجد (instruction) قادر يخرجل ال Translation من
ال Homogenous Translation

>> P = Transl (T)

في بالك T هنا 4×4×41 بعدين

41 element → Cartesian space

$$P = \begin{matrix} 0.4 & 0.2 & 0.0 \\ \vdots & \vdots & \vdots \end{matrix} \downarrow 41$$

بامكانك اعمل (plot) لل P

>> plot (P(1,1), P(1,2))

بعد بعمل plot لل (X) $\xleftarrow{\text{against}}$ (Y)

➤ Steps followed in cartesian space trajectory planning & a comparison with joint space

1- Joint space trajectory planning

initial pose T_i , final pose T_f

"movement in cartesian space not specified"

q_i = p560.ikine(T_i) [1*6]

q_f = p560.ikine(T_f) [1*6]

generate trajectory in joint space

q_i = jtraj (q_i, q_f, length (t))

→ Dimension = length (t) × 6

سواء دي أو Cartesian ال (θ) بتاعة ال 6 هيتغيروا

q_i → T_i الزوايا الى هتدرس

q_f → T_f " " "

XY plane θ_1 شكل الحركة

θ_2, θ_3 → constant

2- Cartesian space trajectory planning

T_i → T_f

move from T_i to T_f in a straight line

بعمله تجيبه ما رح منركز الكدا

T_s = ctraj (T_1, T_2, length (t)) → يعمل ال Trajectory planning

→ series of poses in T (4×4) لل Translation وكمان

q_s = p560.ikine6s (T_s) هو اعطاها لل rotation

وكتب دا ال instruction

يعملها

θ_1, θ_2, θ_3 } variable

ملاحظة

ايه هو الفرق بين q_s , q_j ؟؟؟

قيم (q_s) متخلي الـ origin ثابت والـ rest يتركوا خط مستقيم

➤ Plotting the xy path from the cartesian space planning.

Trajectory planning in cartesian space

1- specify starting pose T_1 4*4
2- specify final pose T_2 4*4
3- generate a time vector
4- T_s = ctraj (T_1 T_2 length)
5- q_s = P560.ikine6s (T_s) 41*6 or length(t) *6
6- p560.plot (q_s)

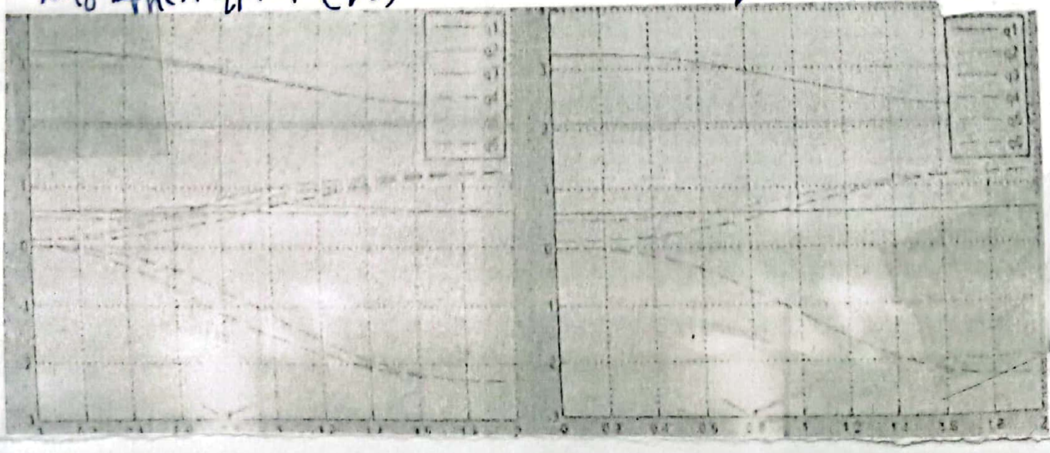
الـ rest هنا تترك في خط مستقيم

➤ Comparing the joint space planning between the two methods.

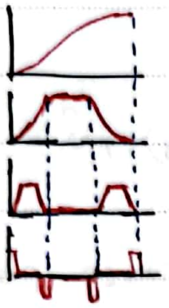
```
>> qplot (t, q)
>> hold
>> qplot (t, qs)
```

والختلاف بين (q, q_s) هو ان الخط او المسار مستقيم في cartesian

وسيرف في joint qplot (t,q) hold then qplot (q_s)



➤ Trajectory planning for a single Axis using Quintic polynomials.



d(t) ~ displacement نه هنا انتقلنا نه نقطة لأخرى وفيه مسار

v(t) ~ This object start with zero speed, finish zero speed

a(t) ~ " " " " " acc, " " acc

j(t) ~ continuous ليه هنا

✓ smooth Trajectory

✓ Temporal derivative → $\frac{d}{dt}$ or $\frac{d^2}{dt^2}$ or $\frac{d^3}{dt^3}$ ———

كل ما كانوا continuous وكل ما كانوا ال Temporal derivative ← continuous

يكون هذا المسار smooth Trajectory

المثال اللي عندي دي ال first & second بس اللي continuous

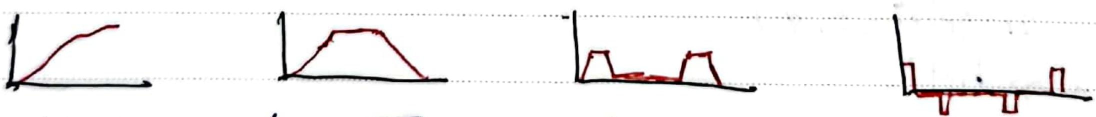
بتاع السرعة والعجلة تحقق

الفكرة دلوقتي أني عاوز أغير النظام بحيث أخلي

d, v, a, j → continuous يكونوا

عشان أحقق ال smooth Trajectory

يعني يكون عندي نعومة في الحركة



(Cubic) (quadratic) linear constant

linear segment فيها   linear فيها جزء

الطريقة اللي بيخدمها في Tool box بيستعمل $\boxed{\text{quintic}}$

و تعتبر polynomial 5 order

$$\theta(t) = At^5 + Bt^4 + Ct^3 + Dt^2 + Et + F$$

displacement $t \longrightarrow 0 : T$

معناه أول displacement هكون بقيمة المهم الفرق بين البداية والنهاية

Conditions:-

1- $\theta(0) = \theta_i$ $\theta(T) = \theta_f$

2- $\dot{\theta}(0) = zero$ $\dot{\theta}(T) = zero$

3- $\ddot{\theta}(0) = zero$ $\ddot{\theta}(T) = zero$

✓ أنا هنا حطيت (٦) شروط ومن المعادلة ال polynomial عندى (٦) متغيرات برضه

✓ هذه الشروط هتتعمل مرة لـ (θ_1) ومرة لـ (θ_2)، وهكذا للكل مع (θ_6)

$$\dot{\theta}(t) = 5At^4 + 4Bt^3 + 3Ct^2 + 2Dt + E$$

$$\ddot{\theta}(t) = 20At^3 + 12Bt^2 + 6Ct + 2D$$

فى الحالة هنا ($T =$ seconds) عشان لو كانت بوحدة تانية هستخدم Chain Rule

if $t \neq$ seconds

$$= \frac{d\theta(t)}{d\Delta t} \cdot \frac{d\Delta t}{dt}$$

دلوقتى عند (٦) معادلات لـ ٦ شروط هنبنى ماتركس ($6*6$)

$$\begin{bmatrix} \theta_i \\ \theta_f \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix}$$

الشروط الأربعة الى بـ zero

$\theta_i, \theta_f \longrightarrow$ بعرفهم من ال (user)

هعوض بـ inverse عشان احسب الباقى A, B, C, D, E, F

- preparing The matrix for solving The coefficients of The quintic polynomials

$$\begin{matrix} \theta(0) = \\ \theta(T) = \\ \dot{\theta}(0) = \\ \\ \\ \\ \end{matrix} \begin{bmatrix} \theta_i \\ \theta_f \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ T^5 & T^4 & T^3 & T^2 & T & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5T^4 & 4T^3 & 3T^2 & 2T & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 20T^3 & 12T^2 & 6T & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix}$$

$\because$ $T, \theta_i, \theta_f \rightarrow$ given

$\therefore$ $A, B, C, D, E, F \rightarrow$ required

by inverse matrix solve and get required.

- solving The matrix in Matlab & comparing it To The Tpoly function

Video Matlab

```
>> T = 3
dt = 0.1
t = [0 : dt : 3]
>> theta = [Pi/2  Pi  0 0 0 0]
>> A = [        6 * 6         ]
>> X = inv(A) * theta
X =
```

تعريف ال t

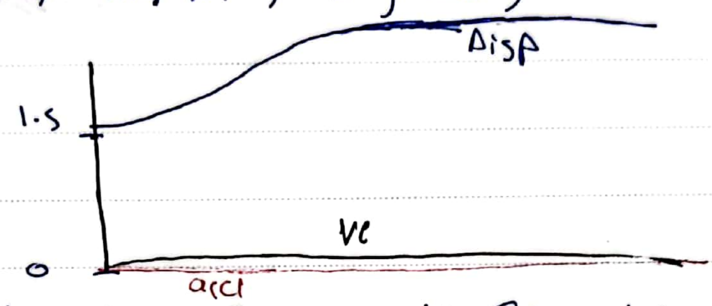
θ_i ↓ Pi/2, θ_f ↓ Pi

Boundary conditions تعريف ال

X =
✓ A
✓ B
✓ C
✓ D
✓ E
✓ F

تحريك ($\ddot{\theta}, \dot{\theta}, \theta$)

```
>> ang_disp = x(1)*t.^5+x(2)*t.^4+x(3)*t.^3+x(4)*
        t.^2+x(5)*t+x(6);
>> ang-vel = dt*(5*x(1)*t.^4+4*x(2)*t.^3+3*x(3)*t.^2+2*x(4)*.
        t+x(5));
>> ang-acc = (dt^2)*(20*x(1)*t.^3+12*x(2)*t.^2+6*x(3)*t+2*x(4));
>> plot(t, ang-disp, t, ang_vel, t, ang_acc)
grid
legend('Ang Disp', 'Ang vel', 'Ang accl')
```



عشان اوضح الكيرڤات استخدم ال Command بتاع ال subplot افضل عشان كل كيرڤ يبقى ليه scale على الرسم خاص بيه

note

١- >> tpoly(θi, θf, length(t))

يسمى quintic polynomial

ال Command ده موجود في Toolbox وده بيحدد نص الكيرڤ بتاع ال plot الفرق بينهم طريقة كتابة الكود بس

٢- الفرق بين tpoly & jTraj

ده single axis (one θ)

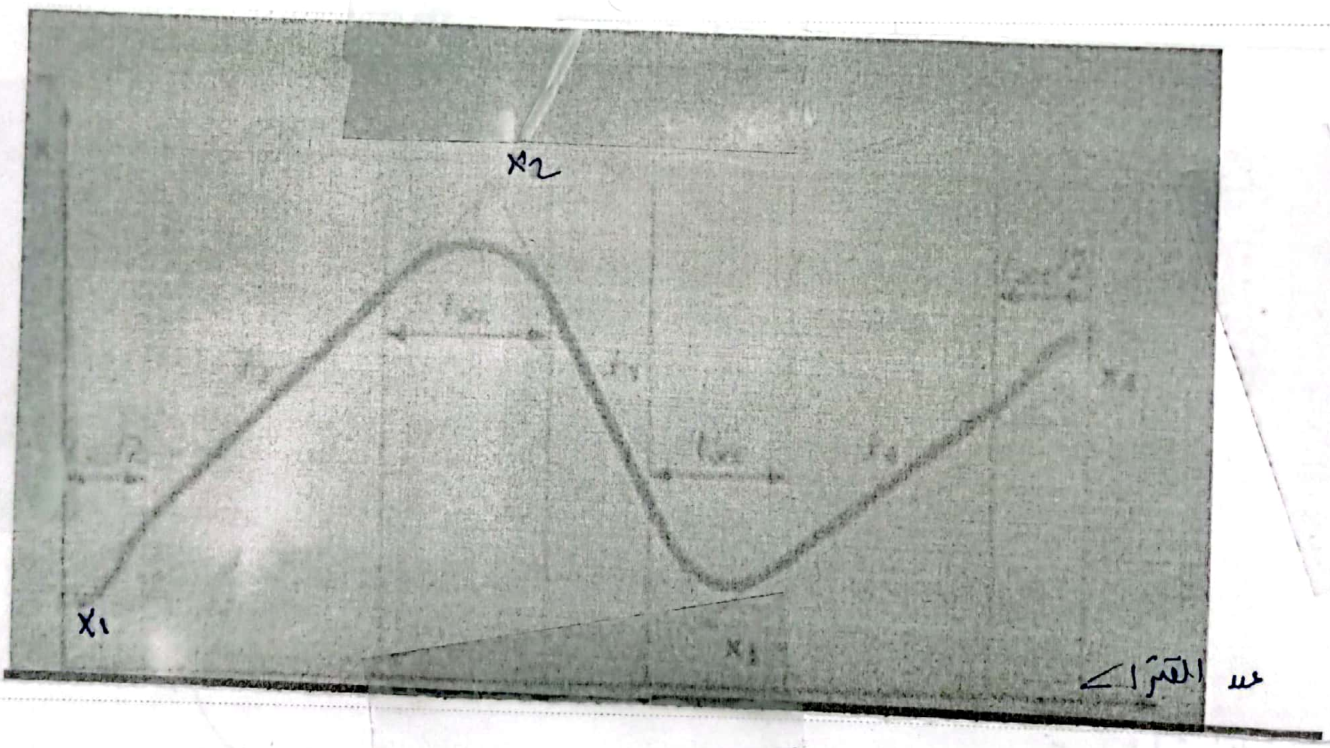
multiple axis

مش لازم يكونه (6) حسب (length q i) (mult θ)

➤ Multi_segment Trajectory planning (mstraj)

Video Matlab

(Mstraj) تشمل العمل Multiple segment
ويختلف منه (j Traj) انه لما (OTraj) الـ velocity
نقط (X1,X2) متكون zero velocity وفيه continuity
للمسار بتاعه
وكمان (MSTraj) تشمل اصد النقاط الـ (in between)
اللي ~~بيمر~~ فيها انحناءات



عند الـ اقتراب

EX

>> Via=[4,1; 4,4; 5,2; 2,5]

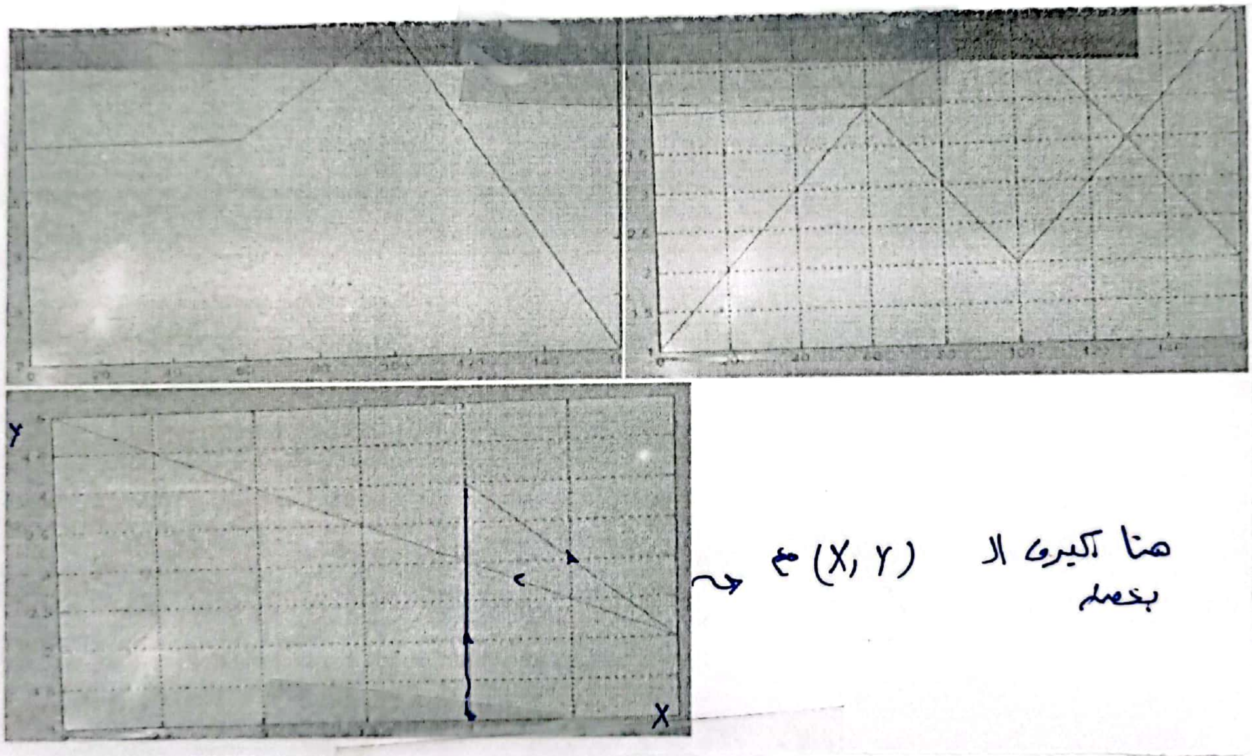
طريقة كتابة الأمر ده mstraj(segmants, QDmax, Q0, DT, Tacc, QD0, QDF, options)

>> q = mstraj(Via, [2,1], [], [4,1], 0.05, 0)

QDmax → max velocity in (X)
      → " " " (Y)

QD0 → initial point   Tacc → Accelaration

```
>> plot (q,:,1))
>> hold on            >> plot (q(1,2))
grid
>> close
plot(q(:,1), q(:,2))
grid
```



هنا آكبرى ال (X, Y) مع
بخصلم

- Two Robotic Toolbox Commands for Translation & Rotation from T

مراجعة

Commands

$T = transl(x, y, z)$

$T = trotX(\theta x)$

$$\begin{bmatrix} \boxed{Rot} & \boxed{Trans} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

EX//

$T_1 = transl(0.4, 0.2, 0)$ ~~* trotx(pi)~~

$$\begin{matrix} 1 & 0 & 0 & 0.4 \\ 0 & 1 & 0 & 0.2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

$T_2 = trotx(Pi)$

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

$T_1 * T_2 =$

$$\begin{matrix} 1 & 0 & 0 & 0.4 \\ 0 & -1 & 0 & 0.2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

➤ joint space Trajectory planning a: Comparison To Cartesian space

Joint space ⟺ Cartesian space

وضعية الربوت معرفة بدلالة T

وضعية الربوت هنا معرفة بدلالة (θ)

eg. RRRRRR

$q = [\theta_1\ \theta_2\ \theta_3\ \theta_4\ \theta_5\ \theta_6]$

$$T = \begin{bmatrix} \boxed{\theta_{x,y,z}} & \begin{matrix} x \\ y \\ z \end{matrix} \\ 0\ 0\ 0 & 1 \end{bmatrix}$$

توضيح الفرق

السؤال الي عاوز اسئله بال (inv.kin)

لو عاوز احقق الوضعية النهائية للـ Tool الي ليها x,y,z و orientation مفروض يدين (q) وممكن ميكونش فيه حلول وممكن أكثر من حل

السؤال التاني 6 (Joint) لو أنا حطيت الربوت مثلا ب θ_1, θ_2, وهكذا ايه الوضعية النهائية

$$q \xrightarrow{f.kin} T$$

$$q \xleftarrow{i.kin} T$$

الأسهل لوصف الربوت ب (q)

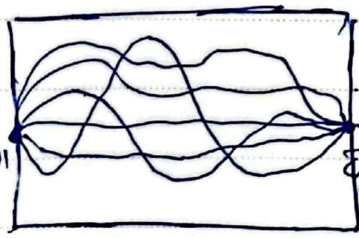
$q = [\theta_1\ \theta_2\ \theta_3\ \theta_4\ \theta_5\ \theta_6]$

waist (θ_1), shoulder (θ_2), elbow (θ_3), rest (θ_4, θ_5, θ_6)

#

q_i, q_f, t_i, t_f } j Traj → joint space

$\theta_1 \to \theta_6$, $\theta_1 \to \theta_6$

البداية  النهاية

البداية والنهاية بتوع الـ واحدة

➤ joint space Trajectory planning b: Finding The initial pose and The Final pose

Video MatLab

```
mdl_puma560
T1=transl(0.4, 0.2, 0)*trotx(pi)
T2=transl(0.4, -0.2, 0)*trotx(pi/2)

q1=p560.ikine6s(T1)
q2=p560.ikine6s(T2)
t=[0:0.05:2]'
q=jtraj(q1, q2, t)
[q, qd, qdd]=jtraj(q1, q2, t)
p560.plot(q1)
p560.plot(q2)
p560.plot(q)
```

50 msec

creation for vector (41 element)

Transpose To make row → column vector

(41×6) by Joint space

>> qplot (t,q)

>> qplot (t,qd)   >> qplot (t,qdd)

Pose -1

T1 =

1.0000	0	0	0.4000
0	-1.0000	-0.0000	0.2000
0	0.0000	-1.0000	0
0	0	0	1.0000

T2 =

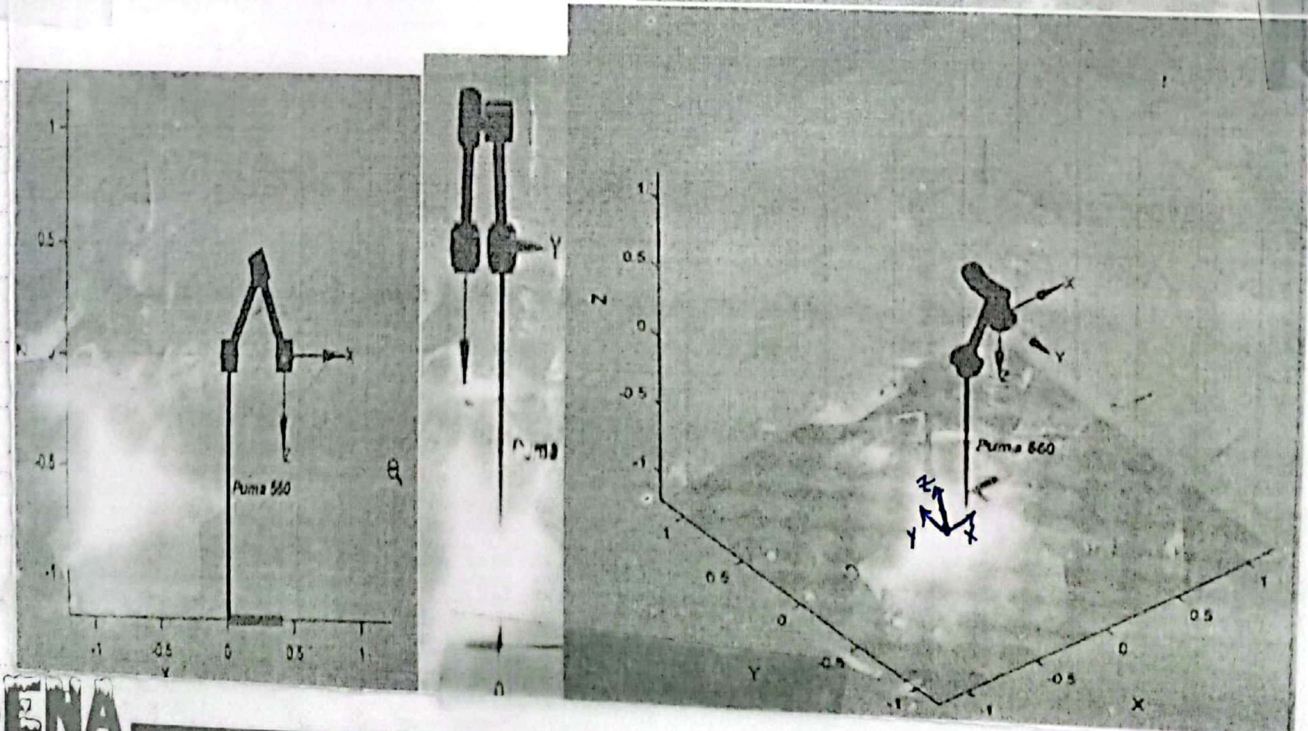
1.0000	0	0	[illegible]
0	0.0000	-1.0000	-0.2000
0	1.0000	0.0000	0
0	0	0	1.0000

q1 =

3.2631	2.0791	0.5992	[illegible]	[illegible]	0.1215

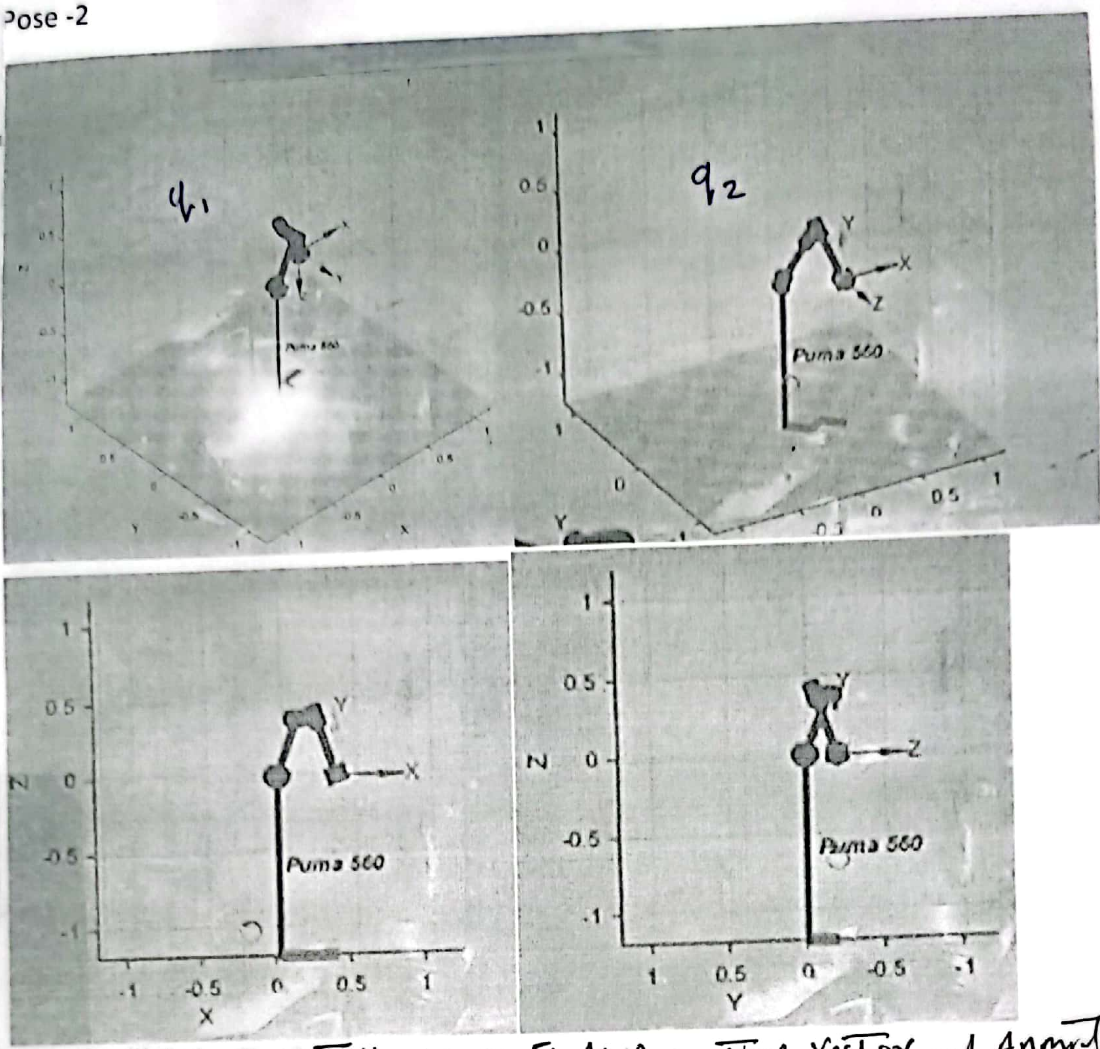
q2 =

2.3358	2.0791	0.5992	-2.3209	1.2425	1.2377



Scanned with CamScanner

الموضوع



- Joint space T. Planning (I Finding a Time vector and Animation

Video Matlab

t	q_i [q_1 ---- q_6]
0	
0.05	
.1	
⋮	
2	q_f [q_{1f} ---- q_{6f}]

$$q_{new} ,\; \begin{array}{c|c|} & 1 \quad\quad 6 \\ 0 & \begin{bmatrix} \\ \\ \\ \end{bmatrix} \\ 40 & \end{array}$$

→ كل element هنا يمثل وضعية في نقطة في الزمن

كل ما أكثر عدد بتاع (Time) يعني لو خليته أكثر مثلا (40) هيدي حركة أكثر smooth للروبوت

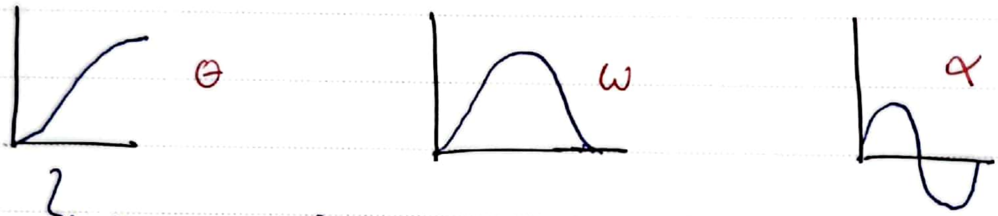
➤ Quintic polynomial for Displacement: plotting Theta, omega and And Alpha of The Trajectory

Video Matlab

quintic = 5 polynomial 5^{th} for $\theta(t)$

>> [q, qd, qdd] = jtraj(q1, q2, q3)

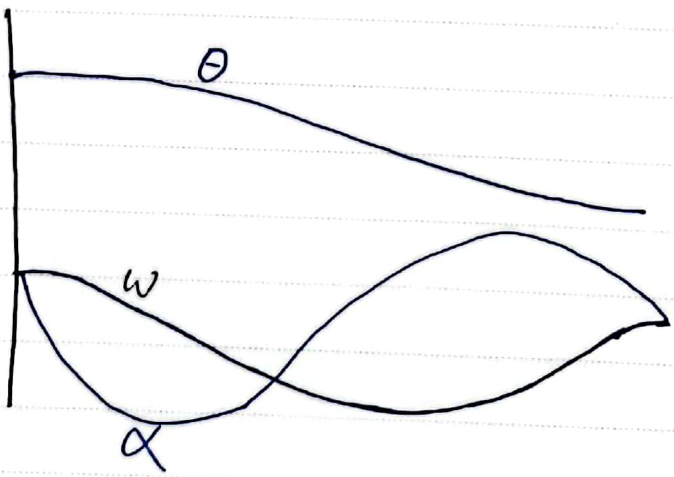
qd		0	0	0	0	0	0	بداية
	41 row							
		0	0	0	0	0	0	نهاية
qdd		0	0	0	0	0	0	بداية
	41 row							
		0	0	0	0	0	0	نهاية

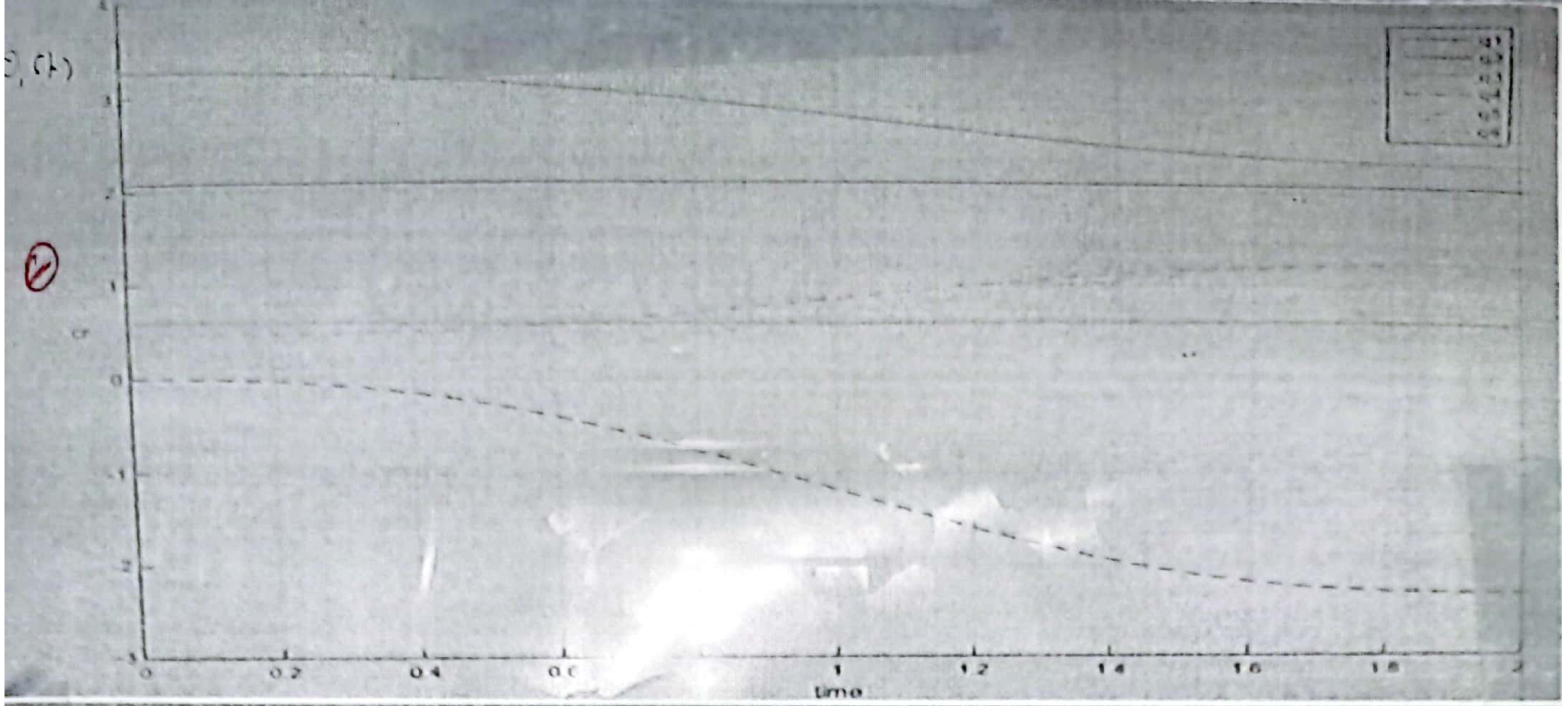


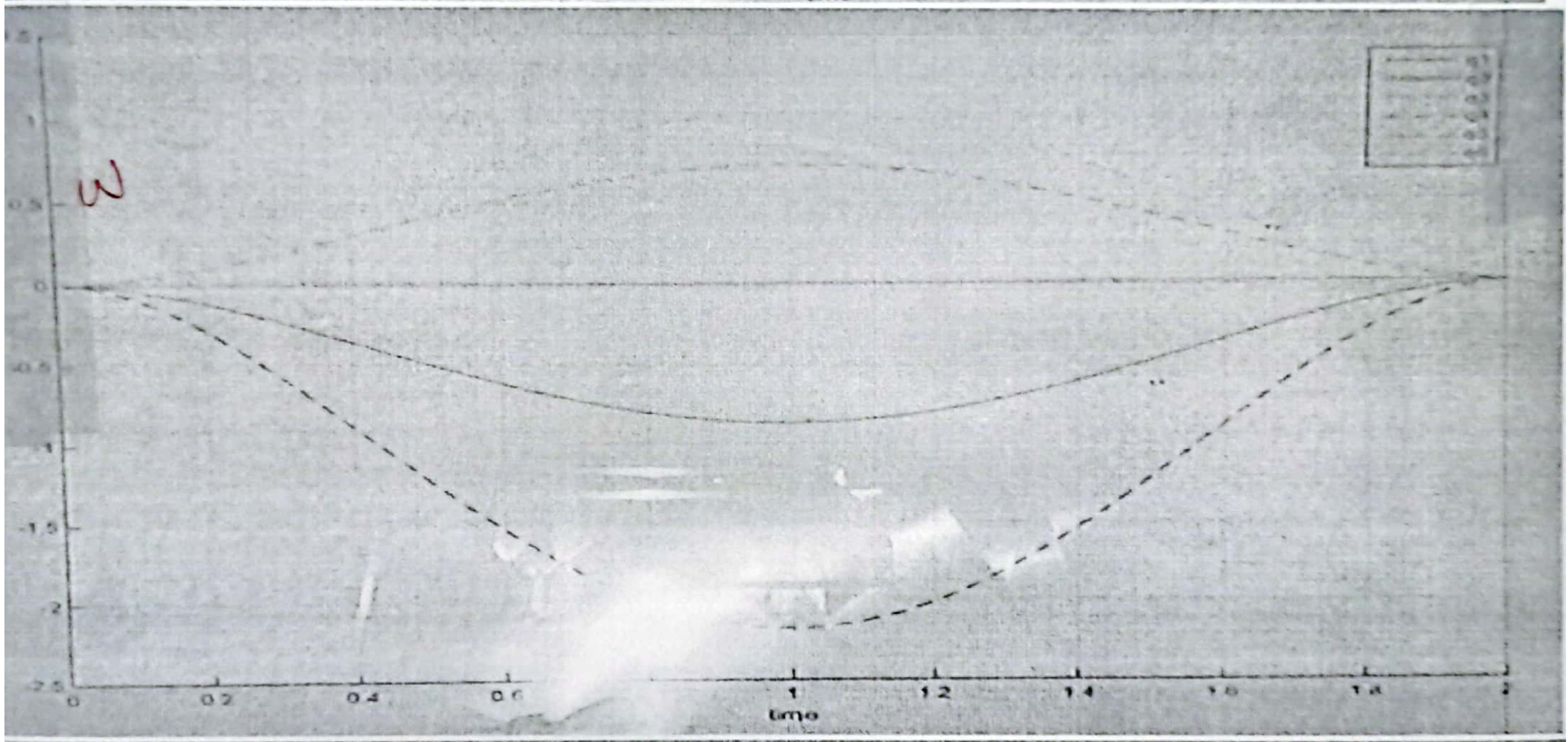
$$\theta(t) = At^5 + Bt^4 + Ct^3 + Dt^2 + Et^1 + F$$

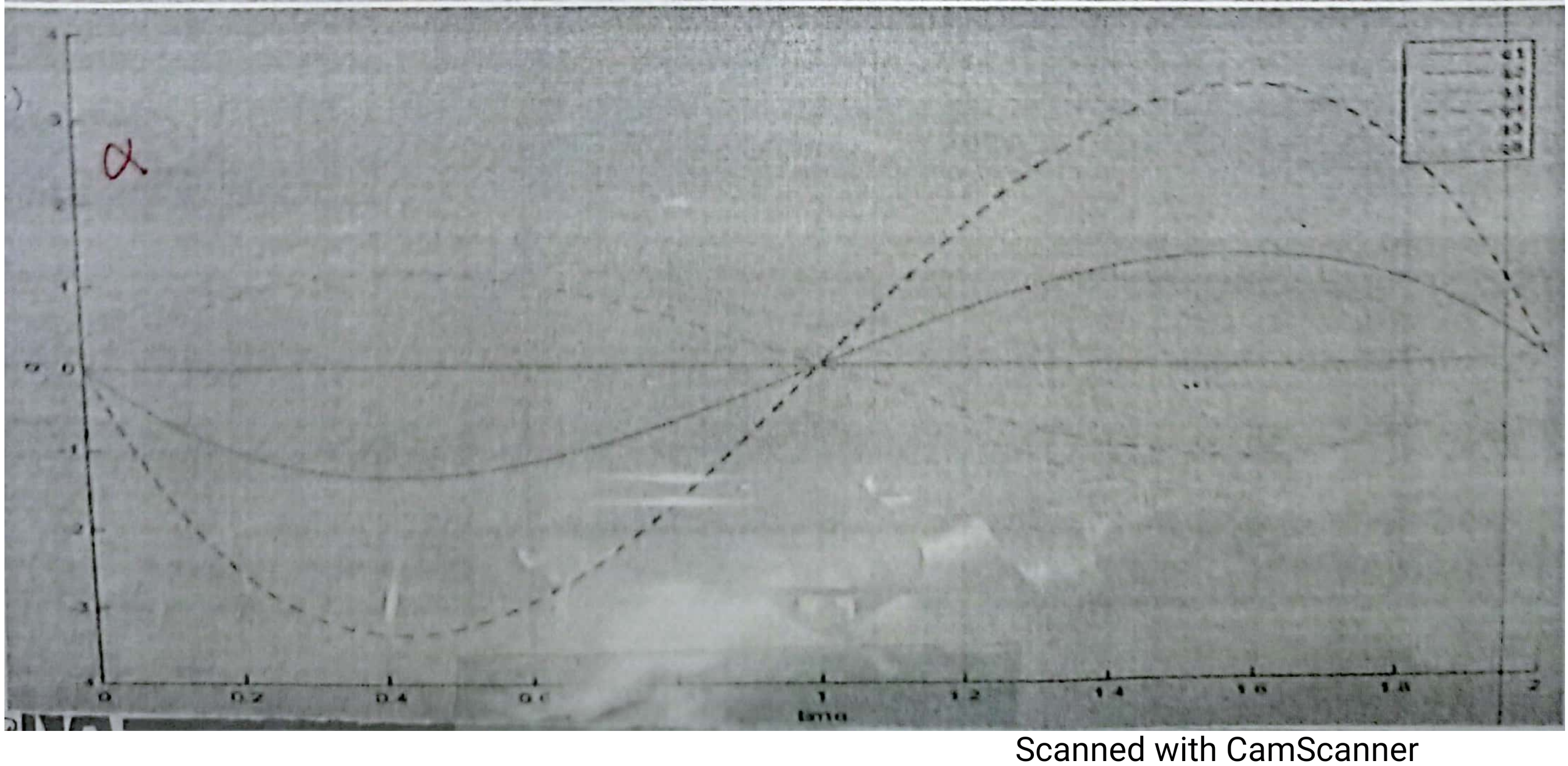
ده (6 Dof) عشان فيه (A, B, C, D, E, F)

(θ_1)




θ
time


ω
time


α
time

➤ Joint space Trajectory planning: plotting The path in The XY Plane

video MatLab

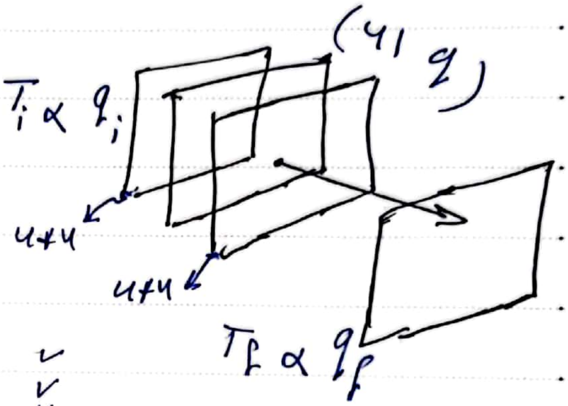
$T_i \Longleftrightarrow q_i$ } → دول طريقتان للتعبير عن وضعية الروبوت

4*4   1*6

>> T = p560.fkine(q) → 41*6

(4*4) → مقدرة (41) مرة

Ans =



T(:, :, 1) = ✓ ✓ ✓ ✓

↓

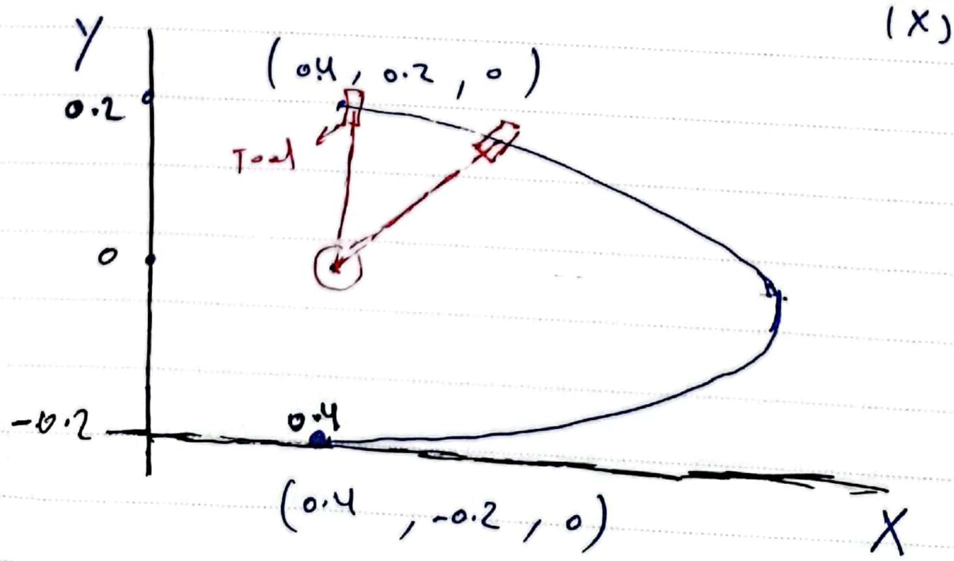
T(:, :, 41)

>> P = Transl(T) → 4*4*41 in space

So, P = ~~4*4*~~ 41 * 3

x ↓ y ↓ z ↓

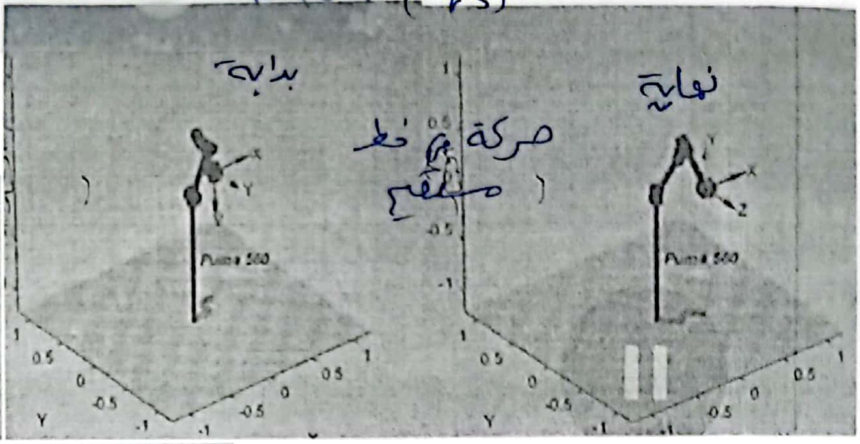
>> Plot(P(:,1), P(:,2)) → plot ل المعاملات column 1 (X), colum 2 (Y)



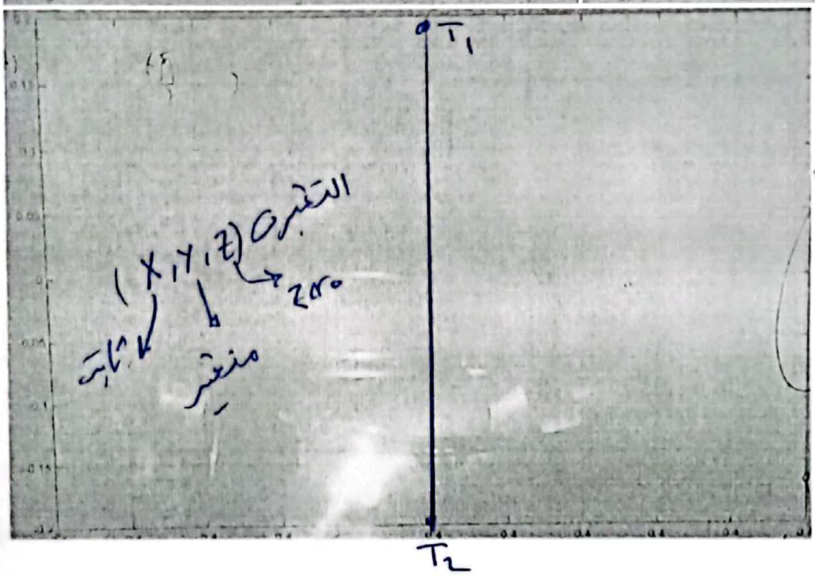
Plan view

- Cartesian space Trajectory planning: plotting the path in the XY plane

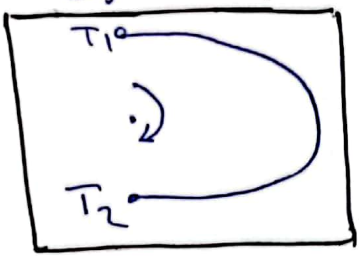
plot(qs)




```
Ts=ctraj(T1,T2, length(t));
qs=p560.ikine6s(Ts)
p560.plot(qs)
ps=transl(Ts)
plot(ps(:,1),ps(:,2))
grid
```



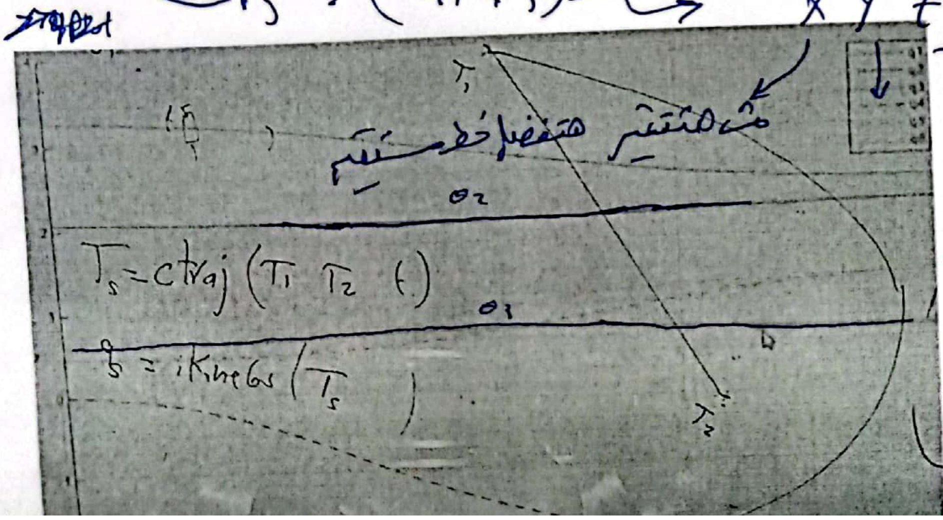
الحالة دي بتحرك من خط مستقيم
الحالة اللي قبلها اتحرك



$T_s = (4*4*41)$

$q_s = (41*6)$

$P_s = (41*3) \rightarrow$ X Y Z التغيرة



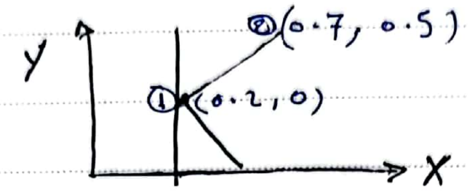
0

>> qplot(t,qs)

حصل هنا تغير لـ
(θ_2, θ_3)

Trajectory planning project: Drawing The letter K(a)

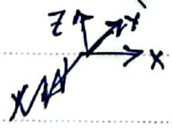
١- الهدف من الكام فيديو دول نرسم حرف (K) في (XY)
مع تثبيت طبعا الحركة في (Z) عندما تكون (Zero)



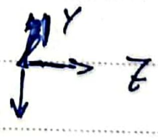
→ plan view

```
>> mdl_Puma560
>> t = [0:0.05:5]
Xi = 0.2     Yi = 0
Xf = 0.7     Yf = 0.5
```

٢- هنفرض أن اتجاه ال (X) لل Puma هو نفسه اتجاه ال Tool
وهلفه حولين ال (Y) بحيث تكون اتجاه ال (Tool) لأسفل
مع فرض أنه طول ال (Tool) يساوي Zero
هنعمل Path planning وماشيين بال Cartesian علشان بمشي من ① لل ②
فاحنا أول حاجة عاوزين نعرف T_i , T_f
فانا عاوز ألفه (90) حولين ال (X) علشان ال (Tool) تبقى لأسفل
لأن الوضع الطبيعي لل Tool



$$T_i = \begin{bmatrix} c\pi/2 & 0 & s\pi/2 & 0.2 \\ 0 & 1 & 0 & 0 \\ -s\pi/2 & 0 & c\pi/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



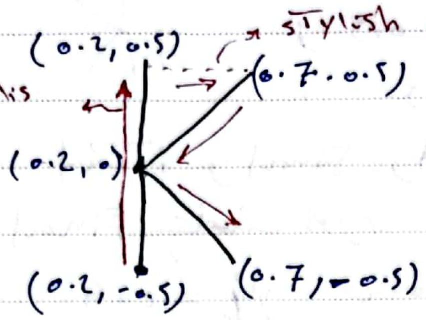
$$T_f = \begin{bmatrix} c\pi/2 & 0 & s\frac{\pi}{2} & 0.7 \\ -s\pi/2 & 1 & c\frac{\pi}{2} & 0.5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
>> T1 = transl(Xi, Yi, 0) * troty(pi/2)
>> T2 = transl(Xf, Yf, 0) * troty(pi/2)
```

➤ Trajectory planning project: Drawing The letter k (b) "Cartesian"

رفع القلم ده stylish off

الأبعاد دي مش بس
رسماني دي مرتبطة
بأبعاد ال Links ه
بتاعت Puma 560



at Z = 0

(X)
X-axis for Tool enter The paper.
حيث ال orientation دائما داخل الصفحة

>> q = iken6s (T)

معناها لو حاوز ~~...~~ احقق ال (T) من Cartesian space
لازم احرك ال Joints بتاعت الروبوت

➤ Trajectory planning project: Drawing The letter k (c)

Video Matlab

الخطوات //

- هكتب البداية والنهاية لكل خط هرسمه وهديله الفترة بسرعة معينة للانتقال من نقطة للأخرى
- المفروض هنا ان ارسم حرف (k) هنادي او هكرر الخطوة الأولى 4 مرات بس الخط التاني مش هرسمه علشان stylish off باطلها بس

كتابة الكود

>> t_2 [0:0.05:5]

>> X_i = 0.2; Y_i = -0.5 ;

X_f = 0.2; Y_f = 0.5 ;

IK

أول مرة هديله initial و final واللي بعدها الـ initial

بتاعها هتبقى الـ Final بتاعت الأولى يعني هتبقى الـ initial

بتاعت الخط التاني Y_f = 0.5 X_f = 0.2

أول واحدة بس لازم ادية الـ final و initial

ملحوظة هامة

انا ماليا شغال Cartesian space لذلك الانتقال بين النقط هيكون على شكل خطوط وليس curves

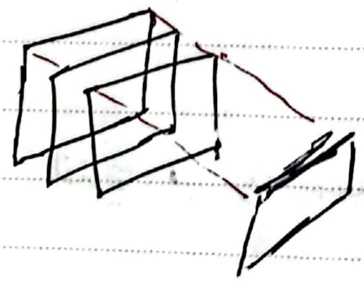
```
>> stylus = true;
```

حركة (Tool) ← يعني لتحت .

```
>> T1 = transl(X1, Yi, 0) * trotY(pi/2)
>> T2 = " (Xf, Yf, 0) * " "
>> T = ctraj(T1, T2, length(t))
>> q = p560.ikine6s(T)
```

(4*4)

vector بعت (lenght*4*4) (101*4*4) (4*4*101)



هنا خد كل (4*4) وطلع منها (q)

اللي هي ايه الـ (pose) اللي هيحقق وضع T_1 و T_2 و T_3

وبالتالي يطلعلي زوايا

```
>> P1 = transl(T1)
>> P2 = transl(T2)
>> X = [P1(1), P2(1)]
>> Y = [P1(2), P2(2)]
>> Z = [P1(3), P2(3)]
```

دا الطبيعي بيرسم هنا الخط اللي Tool مشي عليه أو المسار عشان بيبينلي شكل الحركة في خط مستقيم

الموضوع — التاريخ: / /

```
>> q1 = p560.ikine6s(T1)
>> p560.plot(q1)
>> hold on
>> pause(10);
>> if stylus == True
>>     plot3(x, y, z)
>> end
>> p560.plot(q)
```

هشان بعد أول plot يستنى ١٠ ثواني وينقل على الي بعده

حطيت الشرط ده عشان ميرسمليش الخط رقم اتنين في حرف K

هنا بيمشي في row ورا التاني ويعمل Animation السابقة

$$x_i = x_f \quad ; \quad y_i = y_f$$

$$x_f = 0.7 \quad ; \quad y_f = 0.5$$

K

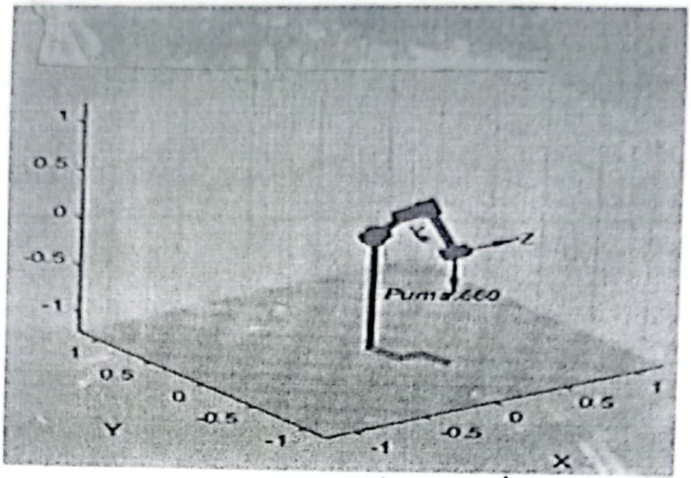
ملحوظة

ممكن اكتب كود ينفذ نفس الخطوات وكله ب function مختلفة أبسط عشان لو عملت تعديل على حاجة هنا لازم اسمك الكود من أوله لآخر اعدله وممكن انسى

هكمل باقي عشان ارسم ال (4) خطوط

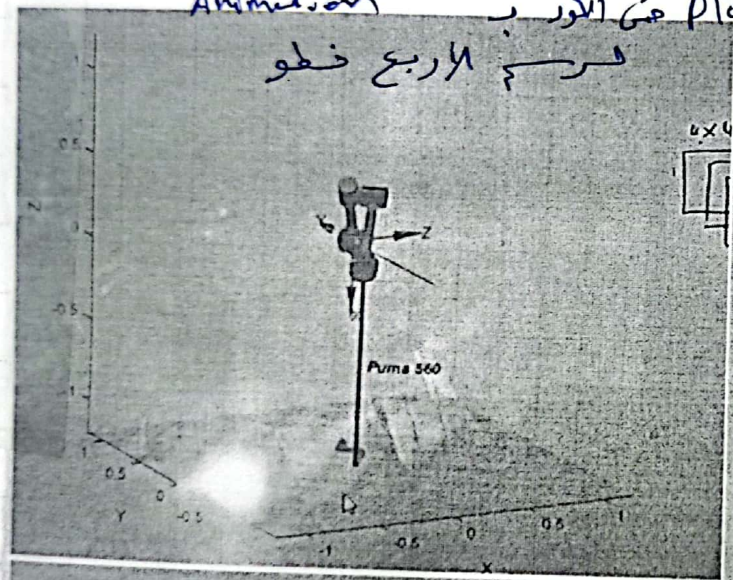
> Trajectory planning project : Drawing The letter K (d)

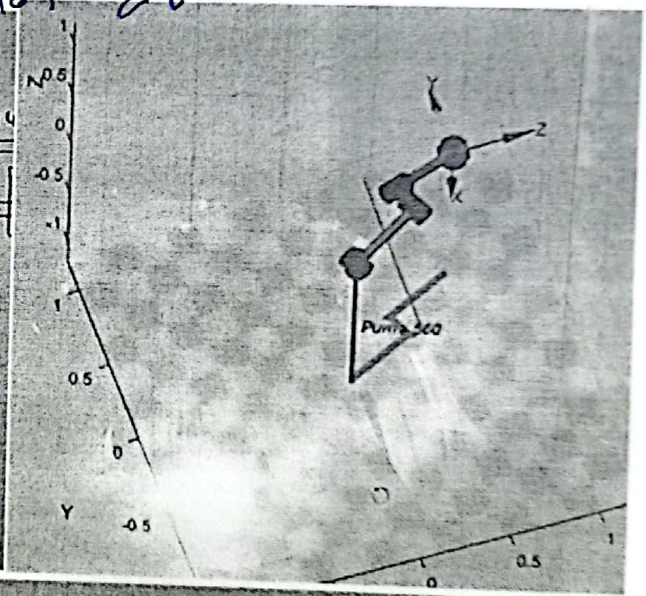
أول plot من الكود

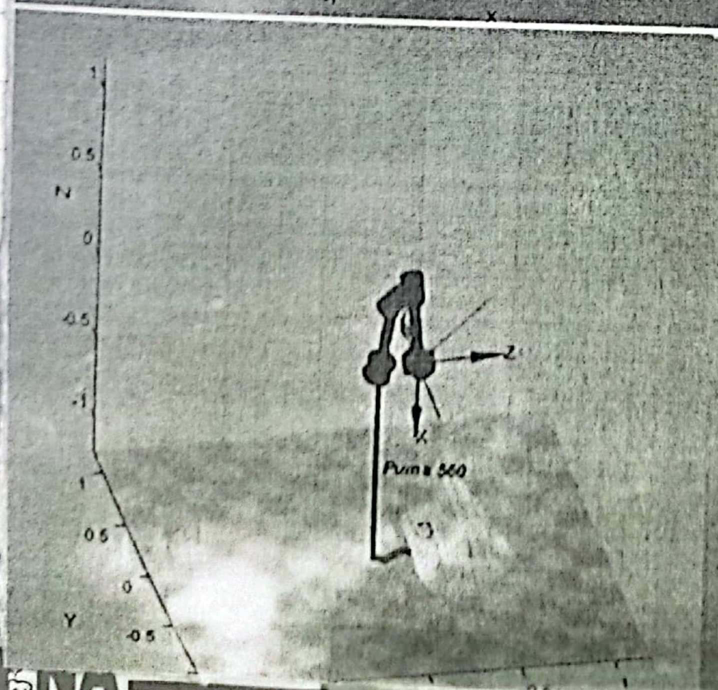


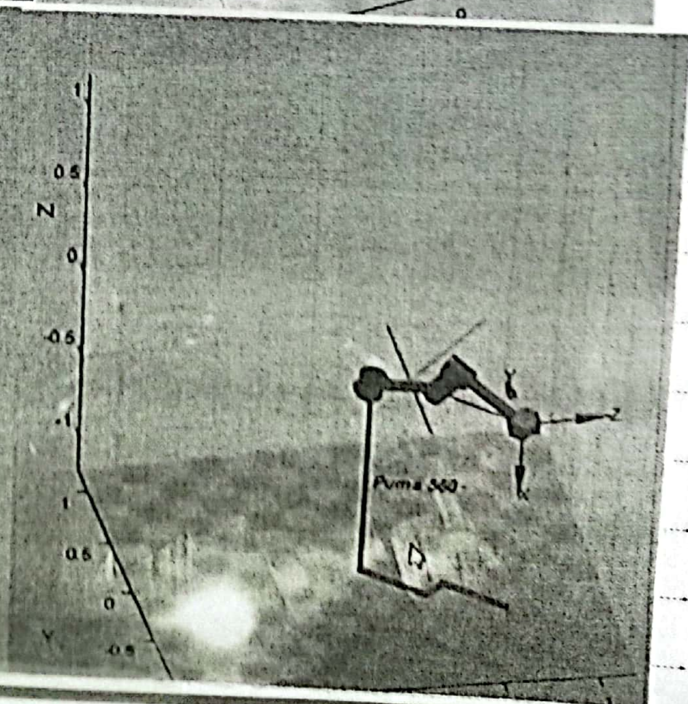
Animation

ثاني plot من الكود ب لرسم الاربع خطو









هاااااااام جدااااا

- Drawing a circle with an RRR Robot
- Drawing a flower
- improvments on The Drawing of The letter k program
- Pick & place App for a Board Game

دول أربع فيديوهات فيهم أكواد ماتلاب لرسم بعض الأشكال
و روبوت التقاط أشياء

- Practical in-lab presentation

فيديو لشرح على الروبوت