

# Tentamen (Python) Programmeren WB1641

Voorbeeld versie (proefversie) gebaseerd op:

Technische Systemen WB1130

Hertentamen MATLAB 11 januari 2017

Omgezet naar Python, 22 + 25 augustus 2017

- Het tentamen duurt **1.5 uur**.
- Werk *netjes en duidelijk*.
- Je kunt in totaal 20 punten verdienen waarvan 2 startpunten.

*Toegestaan op tentamen:*

Kladpapier

*Niet toegestaan op tentamen:*

- Formulebladen wiskunde
- Eigen aantekeningen
- Blokboek of antwoordbladen
- Papieren handleidingen Python of opdrachten Python
- (Grafische) rekenmachine
- Telefoon

Dit tentamen is gemaakt door: Dr. P. Wilders

Dit tentamen is gecontroleerd door: I.A.M Goddijn

## De 'pick & place robotarm'

Figuur A geeft een 'pick & place robotarm' weer. We gaan het bewegingsbereik van de 'pick & place robotarm' bestuderen. Daartoe geeft Figuur B een **schematische weergave** van de 'pick & place robotarm' in zijn werkomgeving van bovenaf gezien.

- De werkomgeving bestaat uit een rechthoekige kamer met dimensies  $A1$  en  $A2$  en oorsprong  $O$
- De pick & place robotarm heeft zijn basis met x y-coördinaten  $(H2, H1)$ , een eerste segment van lengte  $L1$ , een tweede segment van lengte  $L2$  en als end-effector een grijper met basis ter lengte  $2 \cdot L3$  en vingers met lengte  $L3$ .
- De oorsprong van het assenstelsel valt samen met punt  $O$ . De positieve x-richting is naar rechts en deze valt samen met een hoek van 0 graden. De positieve hoeken zijn tegen de klok in gedefinieerd, dus linksom.

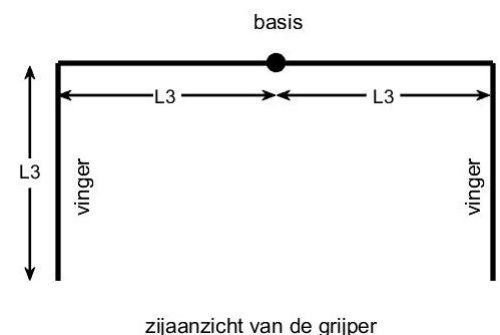
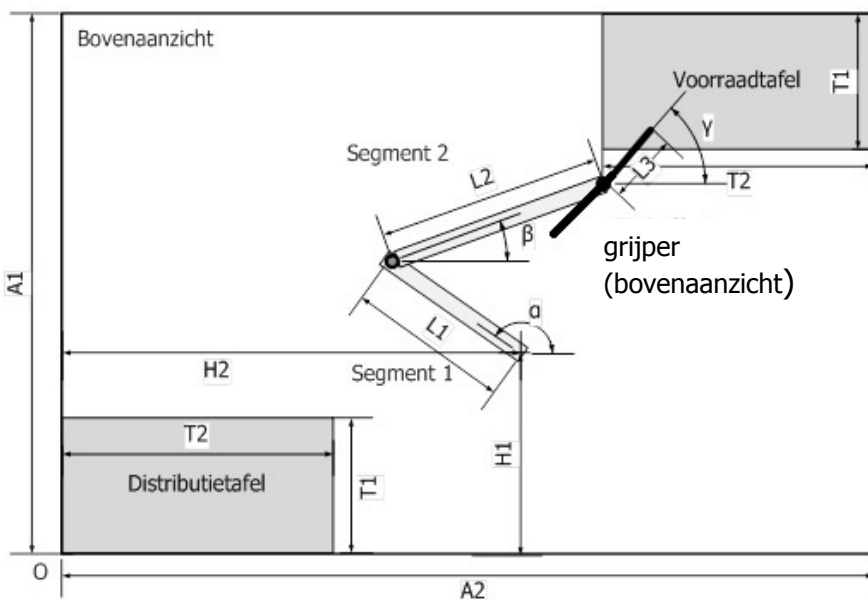


Figuur A. Pick and place robotarm. Bron:

<http://cfnewsads.thomasnet.com/images/large/489/489457.jpg>, View date: aug 2014

Tabel. Waarden van de grootheden die voorkomen in Figuur B.

$A1 = 14 \text{ m}$	$H1 = 0.5 \cdot A1$	$L1 = 5 \text{ m}$	$L3 = 0.2 \cdot L1$
$A2 = 12 \text{ m}$	$H2 = 0.5 \cdot A2$	$L2 = 3 \text{ m}$	
$\alpha = 20 \text{ graden}$	$\beta = 90 \text{ graden}$	$\gamma = 60 \text{ graden}$	$n = 6$



Figuur B. schematische weergave van de pick & place robotarm in zijn werkomgeving gezien van bovenaf.

# Opdracht

Gevraagd wordt een py-file te schrijven die een aantal figuren tekent. Daarnaast moet ook een png-file geproduceerd worden die illustreert hoe lang een bepaald segment kan zijn als aan een zekere conditie voldaan moet zijn.

**Uiteindelijk zal dit resulteren in de gevraagde figuren zoals getoond op de laatste bladzijde(n) van dit tentamen.**

Programmeer de volgende 'stappen' die je op weg helpen om de uiteindelijke Python-code te krijgen in Stap 7. In Stap 8 ga je dan ook nog de code toepassen. Lees elke stap eerst helemaal door en ga deze daarna pas uitvoeren, want de details van de uitvoering komen pas na de eerste zinnen!

## Stap 0 [score: 1]

Tijdens het uitwerken van het tentamen: Programmeer netjes en compact. Zorg dat het programma daadwerkelijk gerund kan worden zonder foutmeldingen. Voeg commentaar (uitleg) toe in de py-file op plekken waarvan u denkt dat toevoeging van commentaar meerwaarde geeft. Gebruik, met mate, lege regels om de overzichtelijkheid te bevorderen.

## Stap 1 [score: 1]

Maak een py-file. Doe dit vanuit de Editor.

- Begin jouw py-file met twee commentaar regels met daarin je naam en studienummer.
- Importeer de modules numpy en matplotlib.pyplot door de opdrachtregels:

```
import numpy as np
import matplotlib.pyplot as plt
```
- toe te voegen.
- Voeg een commentaar regel in met de tekst 'Stap 1'.
- Begin daarna met programmeren door variabelen te maken met namen zoals aangegeven in Figuur B en de Tabel op blz 1 en die de aangegeven parameters bevatten.
- Sla nu de file op als 'WB#####.py' met in plaats van de hekjes je studienummer, dus bijv. WB1234567.py. De tentamenomgeving kiest automatisch een plek voor opslag van het bestand. Het is wel nodig om het bestand de juiste naam te geven. Gedurende het hele tentamen werkt u vanuit dit bestand.

## Stap 2 [score: 2]

Teken de werkomgeving zoals weergegeven in Figuur B. Doe dit als volgt:

- Vervolg met toevoeging van een commentaar regel met de tekst 'Stap 2'. Neem op de volgende regel het commando `plt.figure(1)` op.

- Programmeer nu twee arrays  $Ax$  en  $Ay$  waarin respectievelijk de x- en y-coördinaten komen te staan van de hoekpunten van de werkomgeving op zodanige wijze dat er een complete rechthoek getekend wordt. Maak daarbij gebruik van de variabelen uit Stap 1.
- Programmeer de code om de werkomgeving te tekenen met **dikke groene streep**lijnen.
- Gebruik `plt.axis` om er voor te zorgen dat het assenstelsel in de "scaled" stand wordt gezet. In de scaled stand zijn de lengte eenheden langs de assen gelijk. Het gebruik van "scaled" is te prefereren boven het gebruik van "equal".
- Voeg een titel aan de grafiek toe; 'de pick en place robotarm', en label de x-as met 'afstand X richting [m]' en de y-as met 'afstand Y richting [m]'.
- Zorg ervoor, met gebruik van het commando `plt.axis`, dat de groene rand van de werkomgeving ruim los komt van de assen+box.

Na deze stap moet het programma bij 'runnen' de eerste figuur van de laatste bladzijde(n) tonen.

### Stap 3 [score: 3]

Teken de segmenten  $L1$ ,  $L2$  en de grijper  $L3$  van de **pick & place robotarm** zoals weergegeven in de tweede figuur op de laatste bladzijden.

- Vervolg met toevoeging van een commentaar regel met de tekst 'Stap 3'. Neem op de volgende regel het commando `plt.figure(2)` op om een tweede grafiek te genereren. Ga hierachter door met de introductie van de extra variabelen *alfar*, *betar*, *gammarr* die de waarden van de hoeken alfa, beta, gamma in radialen bevatten.
- Programmeer twee arrays  $Rx1$  en  $Ry1$  waarin respectievelijk de x- en y-coördinaten komen te staan van de twee uiteinden van segment  $L1$ . Doe dat door gebruik te maken van de eerder gedefinieerde variabelen.
- Programmeer twee arrays  $Rx2$  en  $Ry2$  waarin respectievelijk de x- en y-coördinaten komen te staan van twee uiteinden van segment  $L2$ . Doe dat door gebruik te maken van de eerder gedefinieerde variabelen en de arrays  $Rx1$  en  $Ry1$ .
- Programmeer twee arrays  $Rx3$  en  $Ry3$  waarin respectievelijk de x- en y-coördinaten komen te staan van twee uiteinden van de basis van de grijper. Doe dat door gebruik te maken van de eerder gedefinieerde variabelen en de arrays  $Rx2$  en  $Ry2$ .
- Programmeer twee arrays  $Rx4$  en  $Ry4$  waarin respectievelijk de x- en y-coördinaten komen te staan van de twee uiteinden van de eerste vinger van de grijper (projectie op vlak, gedefinieerd door een rotatie van  $90^\circ$  in het  $(y, z)$ -vlak). Doe dat door gebruik te maken van de eerder gedefinieerde variabelen en de arrays  $Rx3$  en  $Ry3$ .
- Herhaal het vorige en programmeer tevens twee arrays  $Rx5$  en  $Ry5$  voor de tweede vinger.
- Programmeer de code om de twee segmenten en de grijper van de pick & place robotarm te tekenen, waarbij de twee segmenten getekend worden met **dikke doorgetrokken zwarte** lijnen en met zwarte 'o'-symbolen aan begin- en eindpunten. Teken de basis van de grijper met een **dikke doorgetrokken zwarte** lijn en beide vingers met **dikke gestippelde zwarte** lijnen (om aan te geven dat het om een projectie gaat).

Na deze stap moet het programma bij 'runnen' (ook) de tweede figuur van de laatste bladzijde(n) tonen.

## Stap 4 [score: 2]

De grijper van de arm kan cirkelvormig bewegen rond het eindpunt van het tweede segment. We vervolgen nu met het programmeren van cirkels. Later zullen cirkels het bewegingsbereik in de werkruimte weergeven.

- Vervolg met toevoeging van een commentaar regel met de tekst 'Stap 4'. Neem op de volgende regel het commando `plt.figure(3)`.
- Maak een nieuw array *gam*, welke een bereik van **0 tot en met 360 graden** met stappen van 4 graden bevat.
- Programmeer daarna de x-coördinaten *Dx* en de y-coördinaten *Dy* van de punten van een cirkel met straal *L3* met behulp van het array *gam* en de variabele *L3* en de functies `np.cos`, `np.sin`. Bij deze programmeerstap ligt het middelpunt van de cirkel in het punt O. Plot deze cirkel met een **dikke doorgetrokken zwarte** lijn.
- Programmeer de arrays *Mx3* en *My3* waarin respectievelijk de x- en y-coördinaten komen te staan van de cirkel getransleerd naar het eindpunt van segment *L2* (d.i. het punt van *L2* waar de grijper aan vast zit). Doe dat door gebruik te maken van (elementen van) de arrays *Rx2* en *Ry2* uit Stap 3. Plot ook deze cirkel met een **dikke doorgetrokken zwarte** lijn.

Na deze stap moet het programma bij 'runnen' (ook) de derde figuur van de laatste bladzijde(n) tonen.

## Stap 5 [score: 1]

- Vervolg met toevoeging van een commentaar regel met de tekst 'Stap 5'. Neem op de volgende regel het commando `plt.figure(4)`.
- Combineer onderdelen uit Stap 2, Stap 3 en Stap 4 en maak in de werkruimte een grafiek van de robotarm in een vaste stand met toevoeging van het bewegingsbereik van de grijper, waarvoor een cirkel met een **dikke gestippelde rode** lijn gebruikt moet worden. Om verwarring te voorkomen rond de weergave van de beweging zien we vanaf nu af van het opnemen van de (projectie van de) vingers van de grijper in de tekening, dus tekenen we de twee segmenten en alleen de basis van de grijper om de arm weer te geven.

Na deze stap moet het programma bij 'runnen' (ook) de vierde figuur van de laatste bladzijde(n) tonen.

## Stap 6 [score: 4]

We gaan nu een figuur maken met erin de robotarm **in *n* (zie Tabel) verschillende standen middels *n* combinaties van hoek  $\alpha$ , hoek  $\beta$  en hoek  $\gamma$ .**

Vervolg met toevoeging van een commentaar regel met de tekst 'Stap 6'. Neem op de volgende regel het commando `plt.figure(5)`.

Teken de werkomgeving zoals in stap 2.

- Programmeer drie arrays *alfa*, *beta* en *gamma*. Array *alfa* heeft een bereik van **20 tot en met 320 graden** en heeft  $n$  elementen die gelijk verdeeld zijn over het bereik. Array *beta* heeft een bereik van **90 tot en met 430 graden** en heeft  $n$  elementen die gelijk verdeeld zijn over het bereik. Array *gamma* heeft een bereik van **60 tot en met 310 graden** en heeft  $n$  elementen die gelijk verdeeld zijn over het bereik.

Bij uitwerking van de volgende vraag kan gekozen worden voor een programmeerstijl die gebaseerd is op het gebruik van *for loops* of er kan gekozen worden voor een programmeerstijl waarin het gebruik van loops voorkomen wordt (*vectorization*). U kunt de stijl kiezen die het beste bij u past, er is geen voorgeschreven stijlfiguur in deze vraag uit Stap 6.

- Programmeer de verschillende standen van de pick & place robotarm door 2D arrays te maken voor segment  $L1$ , segment  $L2$  en de basis van de grijper  $L3$ :
  - Programmeer twee 2D arrays  $Rx1$  en  $Ry1$  ( $n$  rijen, 2 kolommen). Array  $Rx1$  bevat in elke rij de x- coördinaten van de twee uiteinden van segment  $L1$  voor de  $n$  standen.  $Ry1$  bevat in elke rij de y- coördinaten van de twee uiteinden van segment  $L1$  voor de  $n$  standen.
  - Programmeer twee 2D arrays  $Rx2$  en  $Ry2$  ( $n$  rijen, 2 kolommen). Array  $Rx2$  bevat in elke rij de x- coördinaten van de twee uiteinden van segment  $L2$  voor de  $n$  standen.  $Ry2$  bevat in elke rij de y- coördinaten van de twee uiteinden van segment  $L2$  voor de  $n$  standen.
  - Programmeer twee 2D arrays  $Rx3$  en  $Ry3$  ( $n$  rijen, 2 kolommen) . Array  $Rx3$  bevat in elke rij de x- coördinaten van de twee uiteinden van de basis van de grijper voor de  $n$ .  $Ry3$  bevat in elke rij de y- coördinaten van de twee uiteinden van de basis van de grijper.

De genoemde arrays zouden nu volledig gevuld moeten zijn. Pas hierna gaan we de standen ook tekenen.

- Programmeer de code om de verschillende standen van de pick & place robotarm te tekenen. Programmeer hiertoe een **For Loop** waarbinnen de robotarm in de verschillende posities één voor één getekend worden. Hierbij dienen de twee segmenten van de robotarm getekend te worden met **dikke doorgetrokken zwarte** lijnen en met zwarte 'o'-symbolen aan begin- en eindpunten. Teken de basis van de grijper met een **dikke doorgetrokken zwarte** lijn.

## Stap 7 [score: 2]

Vervolg met toevoeging van een commentaar regel met de tekst 'Stap 7'.

Teken in de vijfde figuur nu ook het bewegingsbereik van de grijper weergegeven door cirkels.

- Ten behoeve van initialisatie, maak twee arrays  $Mx3$  en  $My3$  die beide bestaan uit  $n$  rijen en evenveel kolommen als  $Dx$  en waarbij de arrays gevuld zijn met alleen maar **nullen**.
- In  $Mx3$  komen de waarden van de x-coördinaten van de  $n$  cirkels die het bewegingsbereik van de grijper weergeven en in  $My3$  komen de waarden van de y-coördinaten van de  $n$  cirkels die het bewegingsbereik van de grijper weergeven. Zo zal de **eerste** rij van array  $Mx3$  gevuld zijn met de **x-coördinaten** van cirkel  $Dx$  die getransleerd zijn naar de **eerste positie van het**

**eindpunt van segment L2** en de **eerste** rij van array  $My3$  gevuld zijn met de **y-coördinaten** van cirkel  $Dy$  die getransleerd zijn naar de **eerste positie van het eindpunt van segment L2**. De **tweede** rij van array  $Mx3$  wordt gevuld met de **x-coördinaten** van cirkel  $Dx$  die getransleerd zijn naar de **tweede positie van het eindpunt van segment L2**. Etc....

- Gebruik een **For Loop** om de twee arrays  $Mx3$  en  $My3$  te vullen. Maak daarbij gebruik van  $Dx$ ,  $Dy$ ,  $Rx2$ ,  $Ry2$  en richt de for loop zo in dat iedere keer het deel van de arrays corresponderend met één robotarmpositie gevuld wordt.

De genoemde arrays zouden nu volledig gevuld moeten zijn. Pas hierna gaan we de cirkels ook tekenen.

- Programmeer de code om het bewegingsbereik van de grijper  $L3$  ( $n$  cirkels) te tekenen waarbij de cirkels met een '**For Loop** één voor één worden getekend en bestaan uit **dikke gestippelde rode** lijnen.

Na deze stap moet het programma bij 'runnen' (ook) de vijfde figuur van de laatste bladzijde(n) tonen.

## Stap 8 [score: 2]

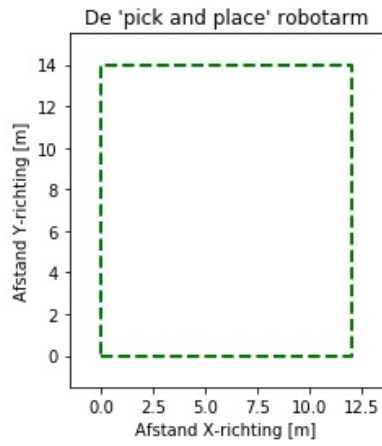
In deze Stap wordt de tot nu toe ontwikkelde code toegepast. Hierbij kan de lengte van segment  $L2$  variëren. Vervolg met toevoeging van een commentaar regel met de tekst 'Stap 8'. Bij beantwoording van deze vraag moet het ontwikkelde Python programma gebruikt worden, beschrijf in commentaar regels in het kort hoe u dit gedaan hebt. Het is dus niet de bedoeling om de vraag te beantwoorden op basis van (alleen) formules.

Zoals geobserveerd kan worden vallen niet alle onderzochte standen van de robotarm en bewegingen van de grijper binnen de werkomgeving. Het segment  $L2$  kan wat ingekort worden. Bepaal de maximale lengte van segment  $L2$  zodanig dat alle standen en bewegingen uit de Stappen 6 en 7 wel binnen de werkomgeving vallen.

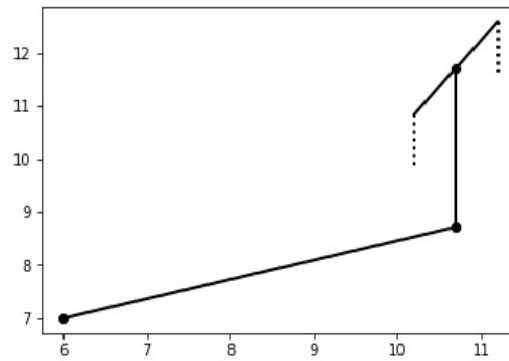
Als antwoord verwachten wij een png-file 'WB#####.png' met in plaats van de hekjes je studienummer, dus bijv. WB1234567.png. Maak deze file vanuit het console door met de rechtermuisknop te klikken op de basisfiguur bevat dat uw antwoord illustreert. Kies voor Save Image as. De tentamenomgeving kiest automatisch een plek voor opslag van het bestand. Het is wel nodig om het bestand de juiste naam te geven. De gepresenteerde grafiek moet tevens de waarde van  $L2$  bevatten welke volgens u gekozen moet worden om aan de eis te voldoen. Plaats deze waarde met gebruik van het `plt.text` in de linker onderhoek van de grafiek in de vorm  $L2 = *** [m]$  met uw waarde op de plek van de sterren.

Dit was de laatste stap van het tentamen. Sluit Spyder af en log uit op de computer (dus niet afsluiten).

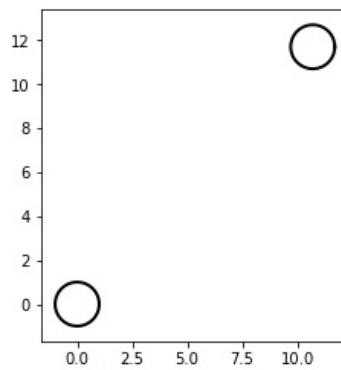
*Zo hoort de eerste  
figuur er uit te zien:*



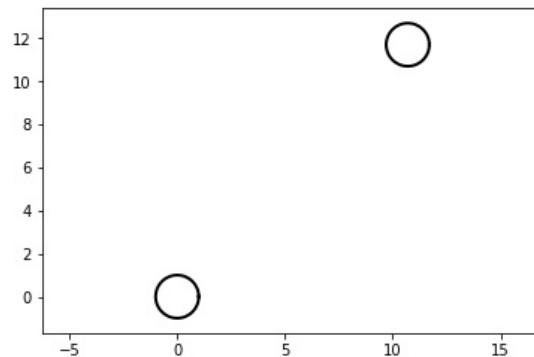
*Zo hoort de tweede  
figuur er uit te zien:*



*Zo hoort de derde  
figuur er uit te zien:*

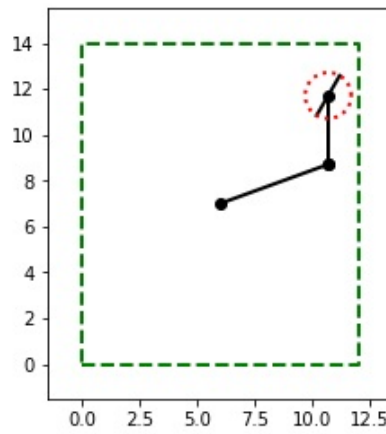


*Of zo:*





*Zo hoort de vierde  
figuur er uit te zien:*



*Zo hoort de vijfde  
figuur er uit te zien:*

