

Tomografie van dynamische systemen

Rewien Durga, 1703447

20 augustus 2018



Universiteit Leiden

Inhoudsopgave

1	Inleiding	3
2	Tomografie	4
2.1	Fourierslice Stelling	4
2.2	ASTRA Toolbox	9
2.3	Dynamische Tomografie	12
3	Stappenplan	14
4	Disk door het vlak	15
4.1	Analytisch	15
4.2	Numeriek	17
4.3	Terugvinden van Parameters (Stap 6)	21
4.3.1	Meetschema (Stap 7)	25
4.4	Willekeurige lijn	25
5	Rotatie van eencelligen	33
5.1	Analytisch	33
5.2	Numeriek	34
5.3	Terugvinden van Parameters (Stap 6)	36
5.3.1	Meetschema	38
6	Conclusie	39
7	Bijlage: Code	40

1 Inleiding

In tomografie worden projecties gebruikt om een reconstructie te krijgen van het binneste van een object. Denk bijvoorbeeld aan het maken van röntgenfoto's. Hierbij wordt van bijvoorbeeld een stilstaand hand, onder verschillende hoeken door een scanner beelden opgenomen. Dit zijn de metingen. Van deze metingen wordt dan middels een reconstructie-algoritme een beeld berekent. Echter zijn objecten waarop tomografie wordt toegepast over het algemeen niet stilstaand maar dynamisch. Dit kan ervoor zorgen dat er meer projecties moeten worden gemaakt om relevante eigenschappen van het object terug te vinden. Wij willen dit aantal minimaliseren. Hiertoe zullen we een aantal dynamische systemen creëren. Dit zijn een aantal tweedimensionale gevallen waarbij we een object door het Euclidisch vlak laten bewegen. Op deze systemen passen we tomografie toe. Het is ons doel om met zo min mogelijk projecties een aantal parameters van het dynamisch object terug te vinden. Hiertoe zullen we een stappenplan maken wat ons het aantal projecties laat minimaliseren. We zullen de projecties zowel analytisch als numeriek via de ASTRA Toolbox^{[2][3]} in MATLAB verkrijgen. Voordat we dit doen, zullen we eerst wat vertellen over de Fourierslice stelling wat ten grondslag ligt aan de tomografie.

2 Tomografie

2.1 Fourierslice Stelling

Centraal in de tomografie staat de Fourierslice Stelling. Voordat we deze geven, hebben we eerst een aantal begrippen nodig. We beschrijven een object in het Euclidische vlak met een functie $f(x, y)$. Van deze functie definiëren we de fouriertransformatie als volgt:

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp(-i2\pi(ux + vy)) dx dy$$

waarbij $(u, v) \in \mathbb{R}^2$. Vanuit $F(u, v)$ kunnen we teruggaan naar het oorspronkelijke object $f(x, y)$ door op F de inverse fouriertransformatie uit te voeren. Deze is als volgt:

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) \exp(i2\pi(ux + vy)) du dv$$

We merken op dat bovenstaande identiteiten alleen gelden als f glad genoeg is en hard genoeg naar nul gaat voor $|(x, y)| \rightarrow \infty$.

Radontransformatie

De Fourierslice stelling zegt iets over de radontransformatie. Deze gebruiken we om te bekijken of bepaalde parameters van het dynamisch object terug kunnen worden gevonden. De radontransformatie is de projectie onder een hoek θ . Een lijn L maakt een hoek θ met de x -as. Hierop staan loodrecht parallelle lijnen $l_{z,x,y,\theta}$. Langs deze lijnen wordt de projectie gemaakt. Hierbij loopt z over L en zijn de lijnen gedefiniëerd als volgt:

$$\begin{aligned} l_{z,x,\theta} &= z \\ &= x \cos(\theta) + y \sin(\theta) \end{aligned}$$

Deze manier van omschrijven van de projectie wat hoort bij een hoek θ , geeft ons de radontransformatie. Deze definiëren we als volgt:

$$\begin{aligned} Rf(z, \theta) &= \int_{l_{z,x,y,\theta}} f ds \\ &= \int_{s=-\infty}^{\infty} f(z \cos \theta - s \sin \theta, z \sin \theta + s \cos \theta) ds \end{aligned}$$

Hierbij is het (z, s) -systeem een rotatie van de originele (x, y) -systeem gegeven door:

$$\begin{pmatrix} z \\ s \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Hierop kunnen we ook de fouriertransformatie toepassen. Dit geeft:

$$S(w, \theta) = \int_{-\infty}^{\infty} Rf(z, \theta) \exp(-i2\pi wz) dz$$

Voorbeeld bij radontransformatie: Schijf rondom de oorsprong

Ter illustratie van wat de radontransformatie doet, bekijken we een voorbeeld^[1]. We nemen een schijf met straal r rondom de oorsprong en passen hier de radontransformatie toe onder verschillende hoeken θ . Een schijf met straal r rondom de oorsprong kunnen we beschrijven met een karakteristieke functie. Deze is 1 in een bepaald gebied en 0 buiten dat gebied. Het gebied wat de schijf omvat noemen we A . De karakteristieke functie die de schijf omschrijft is als volgt:

$$f(x, y) = \begin{cases} 1 & , \text{ als } x^2 + y^2 \leq r^2 \\ 0 & , \text{ anders} \end{cases}$$

Om te achterhalen wat de radontransformatie van f is, merken we op dat de lijn $l_{z,x,y,\theta}$ niet door de schijf A gaat als $|z| > r$. Dit geldt voor alle waarden voor θ . Er geldt dan $f = 0$ op de gehele lijn $l_{z,x,y,\theta}$. Daarmee geldt ook $Rf(z, \theta) = 0$ voor zo een z . Als we echter z bekijken waarvoor $|z| \leq r$, dan zal de lijn $l_{z,x,y,\theta}$ de schijf A doorsnijden langs het deel wat correspondeert aan de parameterwaarden $-\sqrt{r^2 - z^2} \leq s \leq \sqrt{r^2 - z^2}$. De lengte van dit interval is $2\sqrt{r^2 - z^2}$. De waarde van f is 1 op punten in dit interval en 0 elders. Voor een z met $|z| \leq r$ is de waarde van $Rf(z, \theta)$ hetzelfde als dat van de lengte van het gevonden interval, oftewel $2\sqrt{r^2 - z^2}$. We vinden dus:

$$Rf(z, \theta) = \begin{cases} 2\sqrt{r^2 - z^2} & , \text{ als } |z| \leq r \\ 0 & , \text{ anders} \end{cases}$$

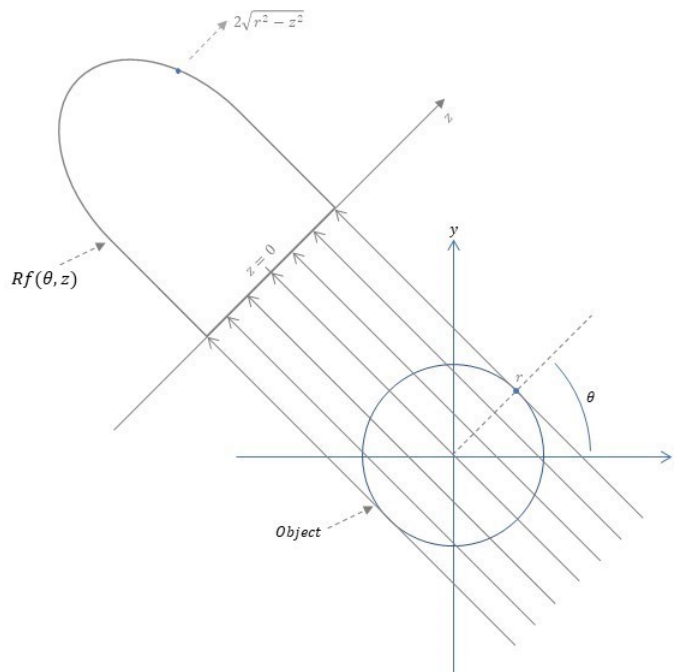
Zie figuur 1 ter verduidelijking.

De fouriertransformatie hiervan is als volgt:

$$\begin{aligned}
S(w, \theta) &= \int_{-\infty}^{\infty} Rf(z, \theta) \exp(-i2\pi wz) dz \\
&= \int_{-r}^r 2\sqrt{r^2 - z^2} \exp(-i2\pi wz) dz \\
&= \frac{r}{w} J_1(2\pi rw)
\end{aligned}$$

Hierbij is $J_1(2\pi rw)$ een Besselfunctie. Besselfuncties $J_n(x)$ zijn oplossingen van de differentiaalvergelijking:

$$x^2 \frac{d^2 y}{dx^2} + x \frac{dy}{dx} + (x^2 - n^2)y = 0$$



Figuur 1: Toelichting bij Radontransformatie van een disk rondom de oorsprong met straal r .

Nu hebben we al het nodige om de fourierslice stelling te formuleren.

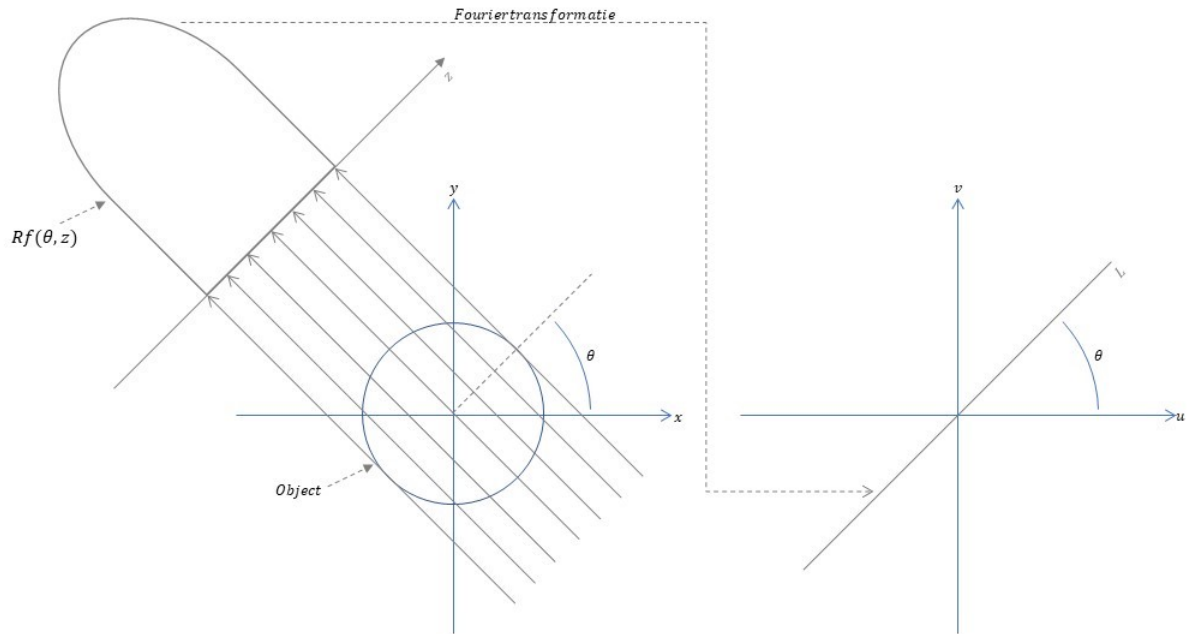
Stelling: Fourierslice Stelling

$$S(w, \theta) = F(u(w, \theta), v(w, \theta)) = F(w, \theta)$$

Als we de fouriertransformatie nemen van een parallelle projectie van een object $f(x, y)$ onder een hoek θ , dan is die hetzelfde als een plak van de fouriertransformatie van het object

waarbij de plak door de oorsprong gaat onder een hoek θ . In andere woorden, de fouriertransformatie van $Rf(z, \theta)$ geeft de waarden van $F(u, v)$ over de lijn L die een hoek θ maakt met de positieve x -as. Hier is L gedefiniëerd als $L = x \cos(\theta)$.

In figuur 2 zien we de stelling toegepast op de disk om de oorsprong ter verduidelijking.



Figuur 2: Toelichting bij Fourierslice Stelling.

Bewijs

$$\begin{aligned}
 S(w, \theta) &= \int_{-\infty}^{\infty} Rf(z, \theta) \exp(-i2\pi wz) dz \\
 &= \int_{-\infty}^{\infty} \int_{s=-\infty}^{\infty} f(z \cos \theta - s \sin \theta, z \sin \theta + s \cos \theta) \exp(-i2\pi wz) ds dz \\
 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp(-i2\pi w(x \cos(\theta) + y \sin(\theta))) dx dy \\
 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp(-i2\pi(xu + yv)) dx dy \\
 &= F(u(w, \theta), v(w, \theta)) = F(w, \theta)
 \end{aligned}$$

De derde gelijkheid volgt uit hoe we (z, s) hebben gedefiniëerd en de vierde gelijkheid uit het nemen van $u = w \cos(\theta)$ en $v = w \sin(\theta)$. \square

D stelling vertelt ons, dat als we een object willen reconstrueren, we een aantal projecties kunnen nemen over opeenvolgende hoeken θ . Op iedere projectie voeren we de fouriertransformatie uit. De fouriertransformaties worden over elkaar gelegd wat ons een beeld geeft opgebouwd uit 1-dimensionale fouriertransformaties van de projecties. Op dit beeld passen we de inverse fouriertransformatie toe, wat ons een reconstructie van het object geeft.

2.2 ASTRA Toolbox

Data zullen we numeriek verkrijgen met behulp van de ASTRA Toolbox. Dit is een toolbox voor zowel MATLAB en Python waarmee 2D- en 3D- tomografie uitgevoerd kan worden. Wij gebruiken de toolbox in MATLAB. Nu volgt een voorbeeld uit de toolbox, waarmee we een aantal commando's zullen presenteren en toelichten:

```
vol_geom = astra_create_vol_geom(256, 256);
proj_geom = astra_create_proj_geom('parallel', 1.0, 384,
    linspace2(0, pi, 180));

P = phantom(256);
[sinogram_id, sinogram] = astra_create_sino_gpu(P, proj_geom, vol_geom);
figure(1); imshow(P, []);
figure(2);
imshow(sinogram(:, :), []);

astra_mex_data2d('delete', sinogram_id);

sinogram_id = astra_mex_data2d('create', '-sino', proj_geom, sinogram);

rec_id = astra_mex_data2d('create', '-vol', vol_geom);

cfg = astra_struct('SIRT_CUDA');
cfg.ReconstructionDataId = rec_id;
cfg.ProjectionDataId = sinogram_id;

alg_id = astra_mex_algorithm('create', cfg);

astra_mex_algorithm('iterate', alg_id, 150);

rec = astra_mex_data2d('get', rec_id);
figure(3); imshow(rec, []);

astra_mex_algorithm('delete', alg_id);
astra_mex_data2d('delete', rec_id);
astra_mex_data2d('delete', sinogram_id);
```

Bovenstaand voorbeeld creëert een reconstructie van het Shepp-Logan figuur, zie figuur 3.

Dit doet de toolbox aan de hand van "parallele straal GPU code"^[4]. Voordat het figuur gemaakt wordt zien we de commando's

```
vol_geom = astra_create_vol_geom (aantal rijen , aantal kolommen)
```



Figuur 3: Shepp-Logan figuur afkomstig uit de toolbox.

```
proj_geom = astra_create_proj_geom(type, afstand tussen twee stralen,  
aantal stralen, projectiehoeken in radialen)
```

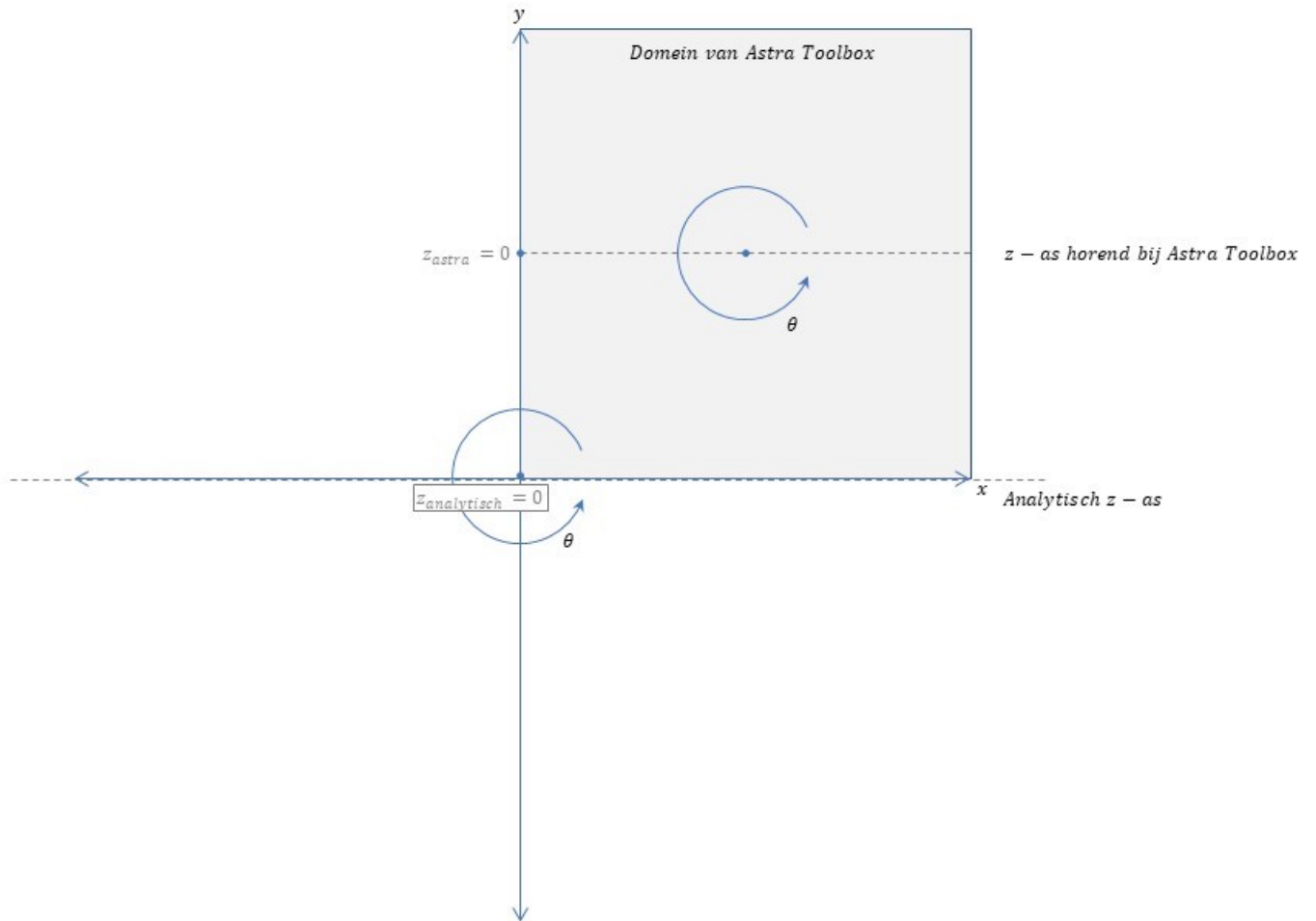
Met de eerste creëren we een 2D volume geometrie. Hierin geven we de afmetingen mee van de projecties. Met de tweede creëren we een 2D type-straal geometrie. Hierin geven we mee wat voor type stralen we gebruiken voor het maken van de projecties, hoeveel pixels er zijn tussen de stralen, hoeveel stralen er gebruikt worden en de hoeken waarover de stralen door het object gaan. We gebruiken parallelle stralen in iedere uitwerking. Vervolgens worden het figuur en de projecties aangemaakt. De projecties maken we aan met de commando

```
[sinogram_id, sinogram] = astra_create_sino_gpu(object, proj_geom,  
vol_geom)
```

De serie projecties wordt weergegeven door *sinogram*. De overige commando's worden ge-

bruikt bij het maken van een reconstructie. Omdat we dit niet doen in ons onderzoek, zullen we de overige commando's niet toelichten.

Wanneer we projecties maken met de ASTRA Toolbox houden we rekening met de werking van de toolbox. Deze Toolbox neemt altijd het centrum van het plaatje als midden van het domein van de projectie en roteert daarom heen, zie figuur 4.



Figuur 4: Positie van de z -as in de Astra Toolbox en analytisch.

2.3 Dynamische Tomografie

We bekijken objecten die bewegen, dus dynamische systemen. Hierop passen we tomografie toe, met als doel terugvinden van parameters die het object beschrijven. De eigenschappen van het object blijven hetzelfde maar het verplaatst wel gedurende de tijd. Op verschillende tijdstippen, onder verschillende hoeken, maken we projecties. Deze zullen we gebruiken in het maken van berekeningen om de parameters te vinden waar we naar op zoek zijn.

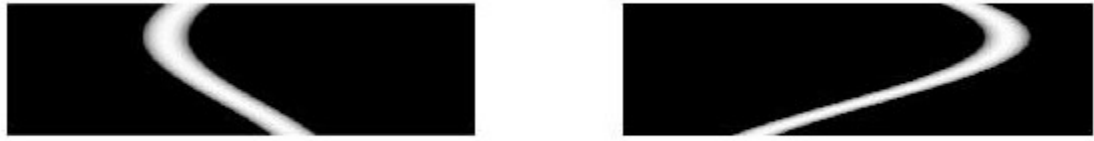
Deze parameters verschillen per systeem. In het hoofdstuk *Disk door het vlak* zullen we een schijf door het Euclidisch vlak laten bewegen langs een lijn. Zie figuur 5 voor een voorbeeld met bijbehorende sinogram in figuur 6.



Figuur 5: 2 momentopnames van een schijf lopend door het euclidisch vlak langs de lijn $y = x$.

Voor het maken van een sinogram worden in dit voorbeeld 180 projecties gemaakt en onder elkaar gezet. Projectie n is hierbij gemaakt onder een hoek $\frac{n}{180}\pi$, met $n \in \{1, 2, \dots, 180\}$. Parameters die het object beschrijven zijn in dit geval de dichtheid ρ en straal r . Parameters die iets vertellen over het systeem zijn de coördinaten van het middelpunt (c_x, c_y) en de snelheid v_x in de x -richting en v_y in de y -richting.

Om antwoord te geven op de hoofdvraag van deze scriptie zullen we een aantal projecties maken op verschillende tijdstippen en onder verschillende hoeken. Per parameter bekijken



Figuur 6: Sinogrammen van de eerdergenoemde momentopnames.

we vervolgens hoeveel projecties we nodig hebben om de parameter terug te vinden. Hierbij gaan we ook na in hoeverre de tijdstippen en de hoeken een rol spelen.

3 Stappenplan

In dit hoofdstuk vertellen we hoe we te werk gaan. Om efficiënt te werk te gaan, houden we een bepaald stappenplan aan. Dit stappenplan geeft een methode voor het terugvinden van de onbekende parameters. De beschrijving is zo algemeen mogelijk, waardoor dit stappenplan toe te passen is bij verschillende dynamische systemen. Er wordt verondersteld dat bekend is hoe de radontransformatie berekend kan worden. Het stappenplan luidt als volgt:

1. Beschrijf het object aan de hand van een functie $f : \mathbb{R}^2 \rightarrow \mathbb{R}, (x, y) \mapsto f(x, y)$ in het vlak.
2. Bepaal de projectie $p : \mathbb{R}^2 \rightarrow \mathbb{R}, (\theta, z) \mapsto p(\theta, z)$ van f onder een hoek $\theta \in [0, 2\pi)$. Dit doen we aan de hand van de radontransformatie.
3. Maak er een tijdsafhankelijk model van. Hierbij maken we k specifieke functies $h_1(a_1, \dots, a_m, t), \dots, h_k(a_1, \dots, a_m, t)$ die k eigenschappen van het systeem beschrijven aan de hand van de m parameters waar we naar op zoek zijn.
4. Beschrijf het dynamisch object aan de hand van een functie $g : \mathbb{R}^3 \rightarrow \mathbb{R}, (x, y, t) \mapsto g(x, y, t)$ met behulp van het tijdsafhankelijk model.
5. Bepaal de projectie $q : \mathbb{R}^3 \rightarrow \mathbb{R}, (\theta, z, t) \mapsto q(\theta, z, t)$ van g onder een hoek $\theta \in [0, 2\pi)$. Dit doen we aan de hand van de radontransformatie en de voorgaande stappen.
6. Vind de parameters terug uit de metingen. In het geval van het object wat langs een lijn beweegt, zouden dat de parameters (c_x, c_y, v_x, v_y) kunnen zijn.
7. Onderzoek onder welke voorwaarden op de metingen de parameters uit de gemeten data kunnen worden bepaald. Hiermee maken we het meetschema. Hierin geven we het minimaal benodigde aantal metingen per parameter weer.

4 Disk door het vlak

We beginnen met het specifieke geval, waarbij we een disk met straal r en dichtheid ρ door het vlak laten bewegen over een lijn. We volgen hierbij het stappenplan. In de paragraaf *Analytisch* volgen we stap 1 tot en met 5. Hierna doen we metingen waarmee we in de paragraaf *Terug vinden van Parameters* stap 6 en 7 kunnen uitvoeren.

4.1 Analytisch

Stap 1

We beginnen met het laten bewegen van een disk door het vlak met een bepaalde snelheid v_x in de x -richting en v_y in de y -richting. De disk met willekeurige middelpunt maken we aan de hand van de volgende functie:

$$f_{c_x, c_y, r}(x, y) = \begin{cases} \rho & , \text{ als } (x - c_x)^2 + (y - c_y)^2 \leq r^2 \\ 0 & , \text{ anders} \end{cases}$$

Dit is een schijf met dichtheid ρ . Voorlopig zetten we ρ vast op 1. De beweging loopt langs de lijn $y = x$. Middels de projectie zullen we proberen enkele parameters terug te vinden. Voorlopig beperken we ons tot de parameters r , v_x , v_y en ρ . We zullen de projectie nu eerst analytisch achterhalen en later numeriek, via de ASTRA Toolbox.

Stap 2

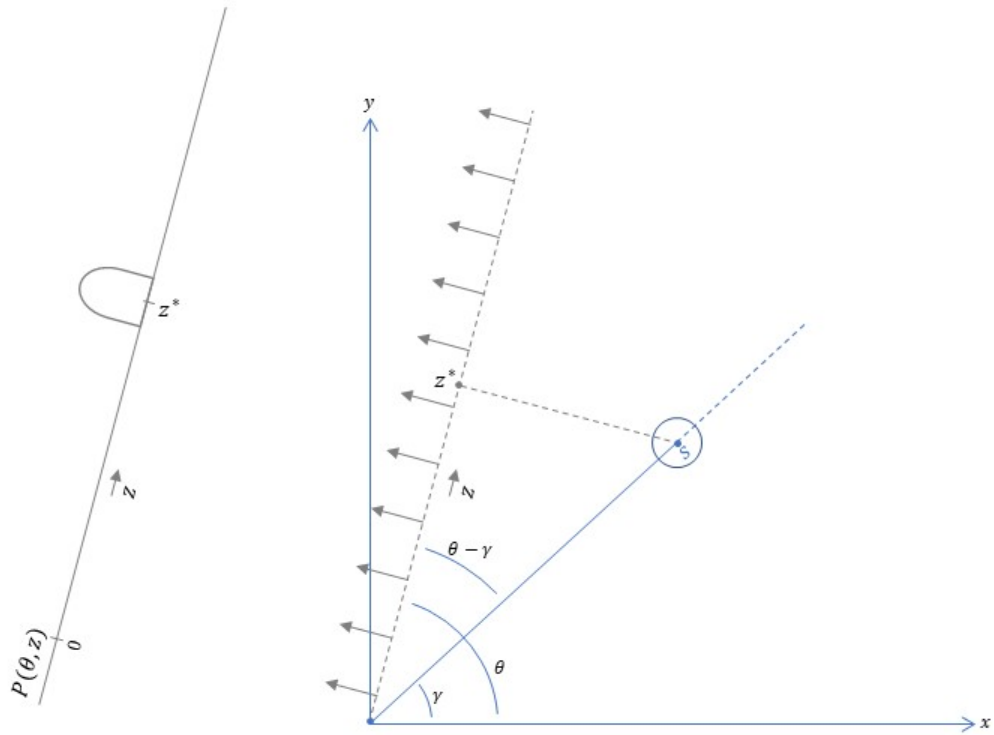
Voor f vinden we een projectie P onder hoek θ conform het eerder besproken voorbeeld. De projectie is nu als volgt:

$$P(\theta, z) = \begin{cases} 2\sqrt{r^2 - (z - s \cos(\theta - \gamma))^2} & , \text{ als } |z - s \cos(\theta - \gamma)| \leq r \\ 0 & , \text{ anders} \end{cases}$$

met $s = \sqrt{c_x^2 + c_y^2}$ en $\gamma = \arctan \frac{c_y}{c_x}$

Dit beschrijft de projectie voor de disk met arbitraire centrum (c_x, c_y) . We leiden deze af van de radontransformatie van de cirkel met straal r en als centrum de oorsprong. Het centrum van de disk wordt in de projectie weergegeven door het maximum van de projectie. Bij het verplaatsen van de disk, verplaatsen we in de projectie het maximum. De afstand van het centrum van de disk naar de oorsprong weergeven we met $s = \sqrt{c_x^2 + c_y^2}$. Beweging vindt plaats over een bepaalde lijn, die een hoek $\gamma = \arctan \frac{c_y}{c_x}$ met de positieve x -as maakt. Met deze gegevens kunnen we berekenen voor welke z^* er een maximum is in de projectie. De projectie wordt over een hoek θ gemaakt. De disk wordt hierbij geprojecteerd op de z -as. In het bijzonder geeft het centrum van de disk, het maximum van de projectie. Dit

maximum is op de z -as bij het punt $z^* = s \cdot \cos(\theta - \gamma)$. Deze is berekend aan de hand van de cosinus-regel. Dit zullen we verduidelijken in figuur 7.



Figuur 7: Toelichting bij achterhalen radontransformatie bewegende disk.

Stap 3

In het geval van beweging langs een lijn, bekijken we de parameters (c_x, c_y, v_x, v_y) . We hebben c_x , het midden van het object als we het zouden projecteren op de x -as en c_y , het midden van het object als we het zouden projecteren op de y -as, beide op tijdstip $t = 0$. We geven het object een bepaalde snelheid v_x in de x -richting en v_y in de y -richting. We nemen $t \in [0, t_e]$ met t_e de eindtijd. Het model ziet er dan uit als volgt:

$$c_x^*(c_x, c_y, v_x, v_y, t) = c_x + tv_x$$

$$c_y^*(c_x, c_y, v_x, v_y, t) = c_y + tv_y$$

Beweging vindt plaats langs de lijn $y = x$, dus er geldt $v_x = v_y$ en $c_x = c_y$.

Stap 4

De gecreëerde dynamische schijf kunnen we beschrijven met de volgende functie:

$$g(x, y, t) = \begin{cases} 1 & , \text{ als } (x - c_x^*(c_x, c_y, v_x, v_y, t))^2 + (y - c_y^*(c_x, c_y, v_x, v_y, t))^2 \leq r^2 \\ 0 & , \text{ anders} \end{cases}$$

Stap 5

De projectie wat hierbij hoort is als volgt:

$$q(\theta, z, t) = \begin{cases} 2\sqrt{r^2 - (z - s^*(t) \cos(\theta - \gamma^*(t)))^2} & , \text{ als } |z - s^*(t) \cos(\theta - \gamma^*(t))| \leq r \\ 0 & , \text{ anders} \end{cases}$$

met $s^*(t) = \sqrt{c_x^*(c_x, c_y, v_x, v_y, t)^2 + c_y^*(c_x, c_y, v_x, v_y, t)^2}$ en $\gamma^*(t) = \arctan\left(\frac{c_y^*(c_x, c_y, v_x, v_y, t)}{c_x^*(c_x, c_y, v_x, v_y, t)}\right)$.

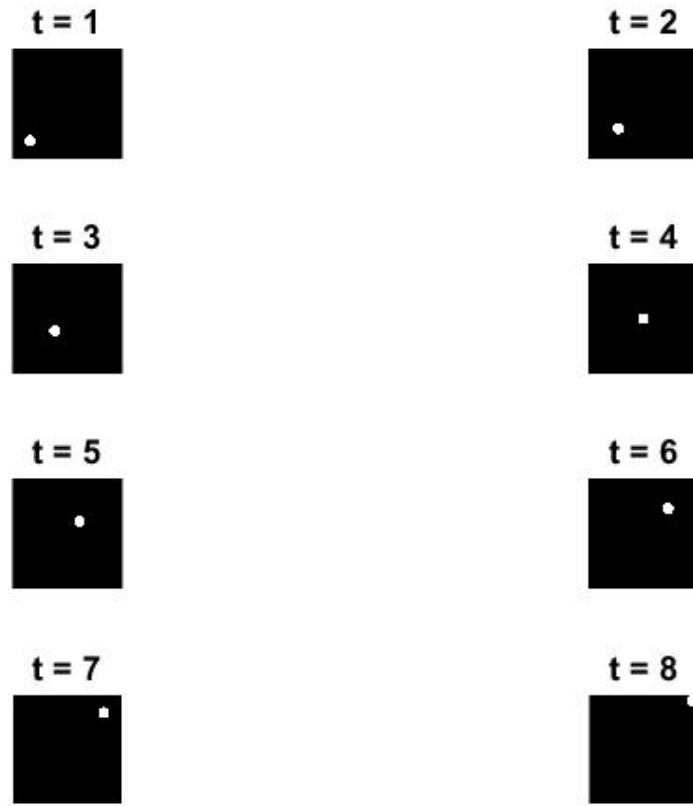
Dit is op precies dezelfde manier afgeleid als voor een vaste centrum.

4.2 Numeriek

We creëren een pixelafbeelding met maten $n \times n$. In deze afbeelding maken we de disk. Omdat we met een pixelafbeelding werken, geeft dit een benadering van een disk met waarde 1 (de dichtheid van de schijf), straal r en middelpunt (c_x, c_y) in het vlak. Voorlopig kiezen we ervoor de disk een straal te laten hebben van $r = \frac{n}{20}$.

We zullen de projectie numeriek vinden via de ASTRA Toolbox. We zullen een vergelijking maken met onze analytische resultaat. We beginnen met een projectie vinden voor $\theta = 0$. Hierna zullen we $\theta \in [0, \frac{7\pi}{4}]$ nemen met stappen van $\frac{\pi}{4}$. Eerst maken we een afbeelding waarin we de positie van de cirkel zien in ons vlak op verschillende tijdstippen. Dit zijn de tijdstippen $t = 1, \dots, 8$. Zie figuur 8.

Voor $\theta = 0$ vinden we figuur 9 via de toolbox. Hiertoe hebben we een programma geschreven wat overeen komt met de code in hoofdstuk 3. Zie de bijlage voor de volledige code. Eerst maken we een afbeelding van 640×640 pixels en beschrijven we het traject, tijdsinterval en de hoeken waarover we projecties maken. Hierna simuleren we een dynamische schijf, waarvan we op verschillende tijdstippen projecties maken. Hierbij zijn de volgende commando's van belang.



Figuur 8: Positie van de cirkel in het vlak op verschillende tijdstippen.

```

vol_geom = astra_create_vol_geom(imagesizeX, imagesizeY)
proj_geom = astra_create_proj_geom('parallel', 1.0, detcount, theta(teller))
[sinogram_id, sinogram] = astra_create_sino_gpu(circle, proj_geom, vol_geom)

```

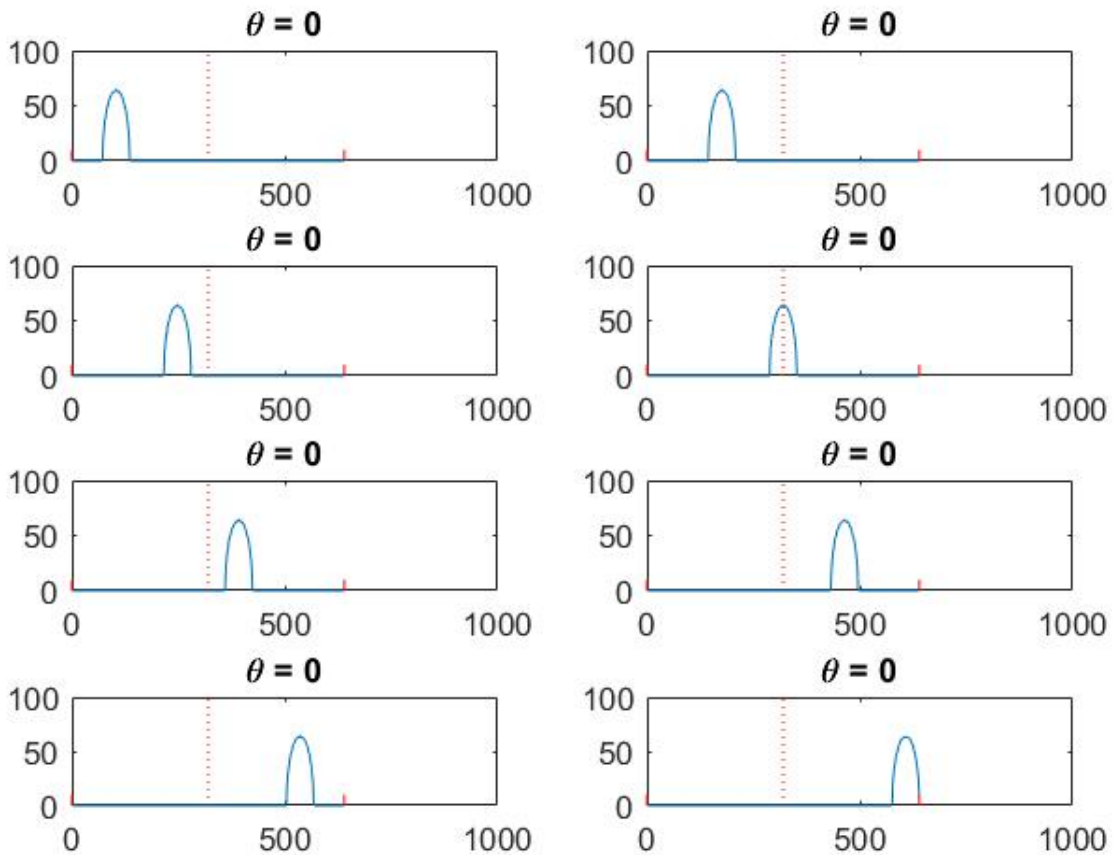
Net als in het voorbeeld maken we eerst een 2D volume geometrie en een 2D parallelle straal geometrie. Hierna creëren we de projecties. Deze plotten we met de volgende commando

```

plot(sinogramx, sinogram, [detcount/2 detcount/2], [0 100], 'r', [0 0], [0 10],
'r', [detcount detcount], [0 10], 'r')

```

Hierin is *sinogramx* het domein van de projectie en geeft *sinogram* de projectie weer.

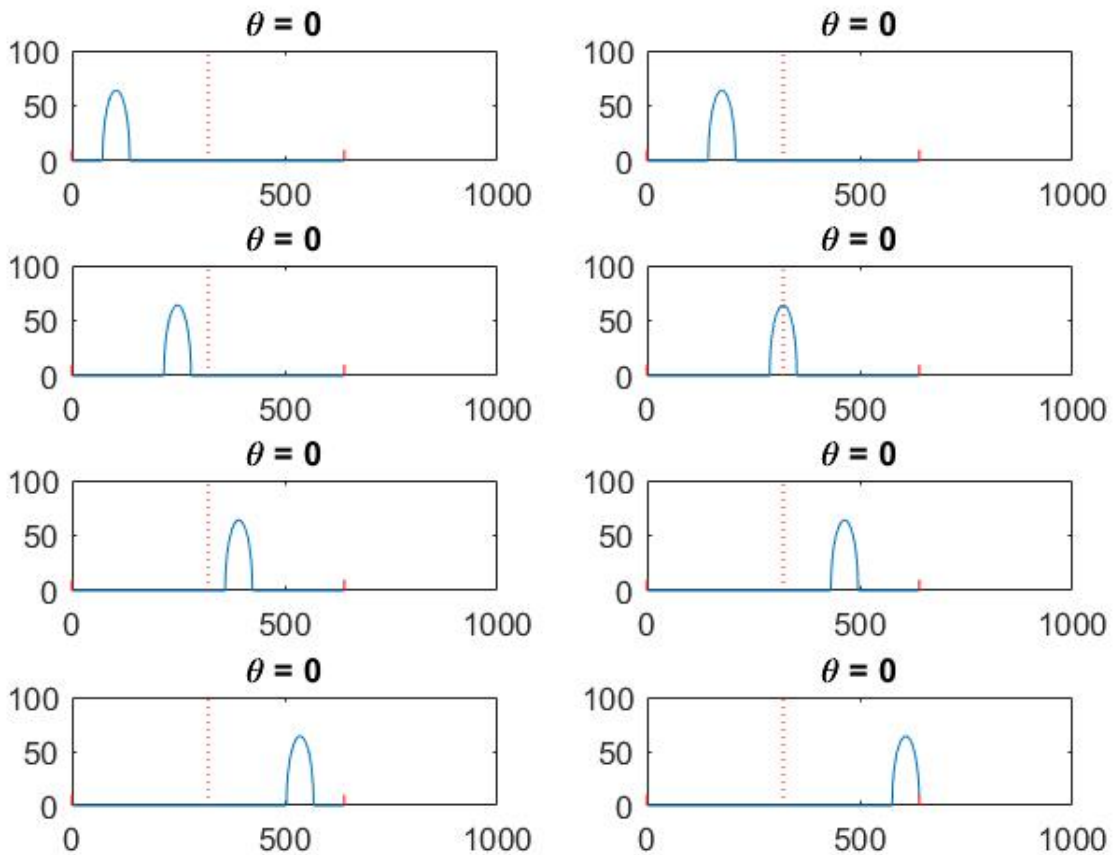


Figuur 9: Projectie (ASTRA Toolbox) van de cirkel door het vlak voor $\theta = 0$ op verschillende tijdstippen.

Boven de verschillende plotjes staat onder welke hoek θ een afbeelding is gemaakt. Het begin en het einde van het domein worden weergegeven door twee verticale lijnen en de stippellijn geeft aan waar het midden is van het domein van de projectie. Als we de formule plotten die we analytisch achterhaalden, dan krijgen we figuur 10. Deze komt netjes overeen met de gevonden projectie via de ASTRA Toolbox.

We zullen het bovenstaande nu uitvoeren voor verschillende θ , met $\theta \in [0, \frac{7\pi}{4}]$ en stappen van $\frac{\pi}{4}$. Via de ASTRA Toolbox vinden we figuur 11.

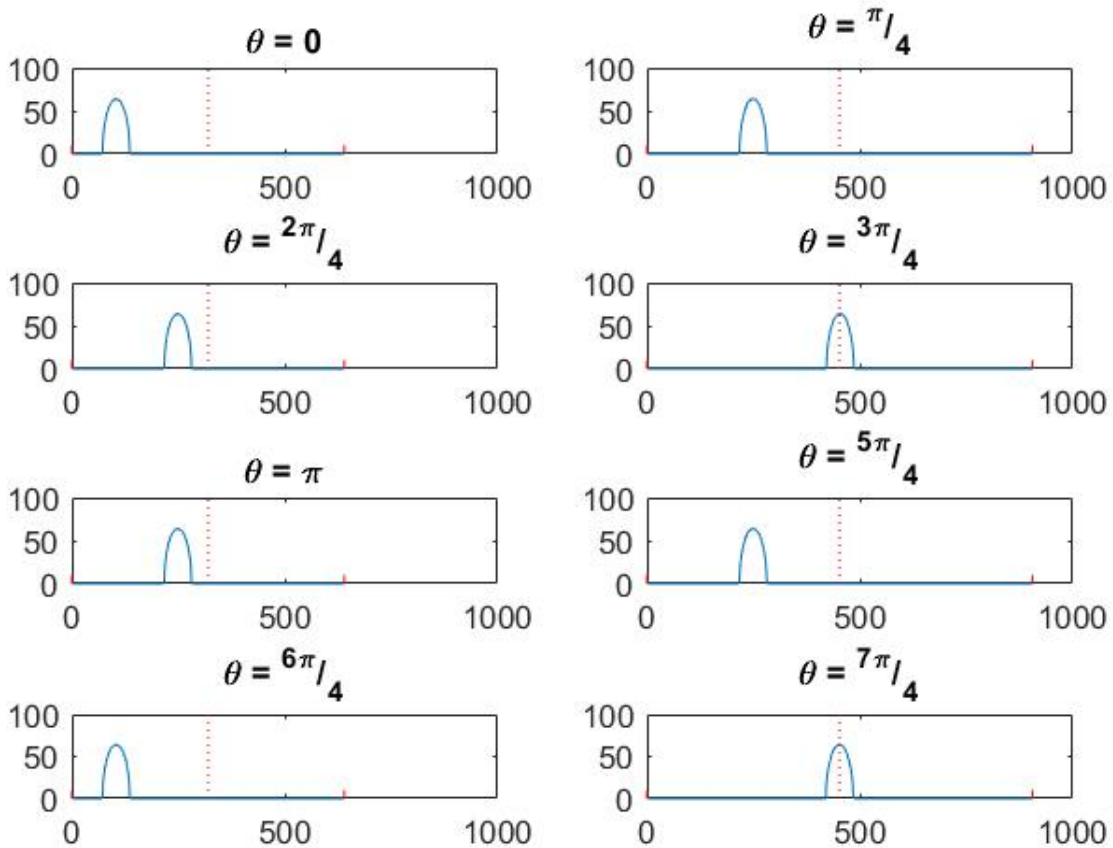
Gegeven dat we met een cirkel werken en langs de lijn $y = x$ lopen is de afbeelding precies zoals verwacht. De afbeelding die we krijgen via de formule gevonden via de analytische



Figuur 10: Projectie (analytisch) van de cirkel door het vlak voor $\theta = 0$ op verschillende tijdstippen.

wijze zien we in figuur 12.

De afbeeldingen komen overeen. Bij de hoeken die veelvoudig zijn van $\frac{\pi}{2}$, kijken we over of de breedte of de lengte van onze afbeelding. In dat geval laten we de lengte van het domein hetzelfde zijn als de lengte (of breedte) van de afbeeldingen. Bij de hoeken $\frac{k\pi}{2} + \frac{\pi}{4}$, $k \in \mathbb{Z}$, kijken we over de diagonalen van de afbeelding. In dat geval laten we de lengte van het domein hetzelfde zijn als de lengte van de diagonaal waarover wordt gekeken. Dit verklaart het verschil in lengte van de domeinen.



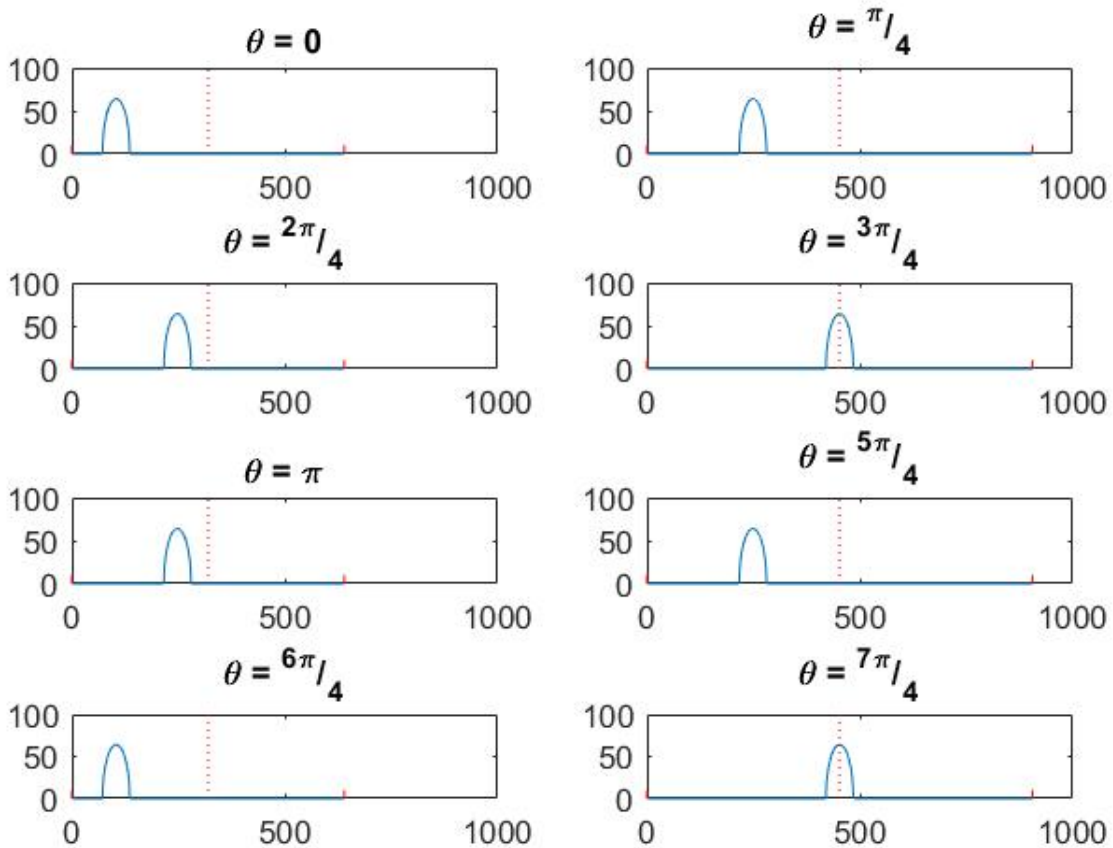
Figuur 11: Projectie (ASTRA Toolbox) van de cirkel door het vlak voor variërende θ op verschillende tijdstippen.

4.3 Terugvinden van Parameters (Stap 6)

We zullen nu de parameters r , v_x , v_y en ρ terugvinden aan de hand van de projecties. Tevens is het een interessante vraag hoeveel projecties onder hoeveel hoeken we nodig hebben om de parameters te vinden.

Het is gegeven dat we een disk hebben. De projectie is voor een gedeelte een halve ellips. De halve ellips vinden we daar waar de stralen gebruikt om de projectie te maken, door de disk gaan. Om de straal r te achterhalen tellen we het aantal pixels waarbij de projectieafbeelding geen waarde 0 heeft. Dit geeft de radius d van de schijf. We berekenen de straal van de schijf via de formule $r = \frac{d}{2}$. Van één projectie tellen we de pixels. Dit geeft de diameter $d = 64$, oftewel $r = \frac{d}{2} = \frac{64}{2} = 32$ pixels.

Voor de snelheid in de x - respectievelijk de y -richting maken we gebruik van de formules



Figuur 12: Projectie (analytisch) van de cirkel door het vlak voor variërende θ op verschillende tijdstippen.

voor het dynamisch centrum:

$$c_x^*(t) = c_x + tv_x$$

$$c_y^*(t) = c_y + tv_y$$

Op verschillende tijdstippen zullen we $c_x^*(t)$ en $c_y^*(t)$ achterhalen. We gebruiken hierbij enkel de projectie. Deze heeft een maximum wat het centrum van de disk weergeeft. De disk beweegt zich langs de lijn $y = x$, dus $\gamma = \frac{\pi}{4}$, $c_x^*(t^*) = c_y^*(t^*)$ en $v_x = v_y$. Het maximum in de projectie staat voor verschillende waarden t^* op:

$$z^*(t^*) = s(t^*) \cos(\theta(t^*) - \frac{\pi}{4})$$

Hierbij is $z^*(t^*)$ de waarde van z bij het maximum in de projectie verkregen uit de me-

tingen. Voor $s(t^*)$ geldt:

$$s(t^*) = \sqrt{c_x^*(t^*)^2 + c_y^*(t^*)^2}$$

voor een $t^* \in [t_0, t_e]$. Gegeven dat $c_x^*(t) = c_y^*(t^*)$ vinden we:

$$s(t^*) = \sqrt{2}c_x^*(t^*).$$

Hieruit volgt dat geldt:

$$z^*(t^*) = \sqrt{2}c_x^*(t^*) \cos(\theta(t^*) - \frac{\pi}{4}).$$

Dit geeft ons:

$$c_x^*(t^*) = c_y^*(t^*) = \frac{z^*(t^*)}{\sqrt{2} \cos(\theta(t^*) - \frac{\pi}{4})}.$$

Deze formule berekent de coördinaten van het centrum van de disk op $t = t^*$ aan de hand van gemeten $z^*(t^*)$ en gegeven $\theta(t^*)$. We merken op dat we hier $\theta(t^*) \neq \frac{3\pi}{4} + k\pi$, $k \in \mathbb{Z}$ eisen. Voor die waarden zal de noemer gelijk zijn aan nul en kunnen we dus niet de positie van de centrum van de disk bepalen. Ook zien we in figuur 13 dat voor $\theta(t^*) \in \{\frac{3\pi}{4}, \frac{7\pi}{4}\}$, het centrum van de disk voor iedere waarde van t^* geldt $z^*(t^*) = 0$. Oftewel, voor willekeurige t^* kunnen we niet de x - en y -coördinaten bepalen van het centrum van de disk voor deze waarden van θ .

Om v_x en v_y te vinden maken we gebruik van de welbekende formule $v_x = \frac{\Delta c_x^*(t)}{\Delta t}$. Omdat we over een lijn bewegen hebben we twee verschillende tijdstippen nodig. We nemen de tijdstippen t_0 en t_1 . Hiervoor vinden we:

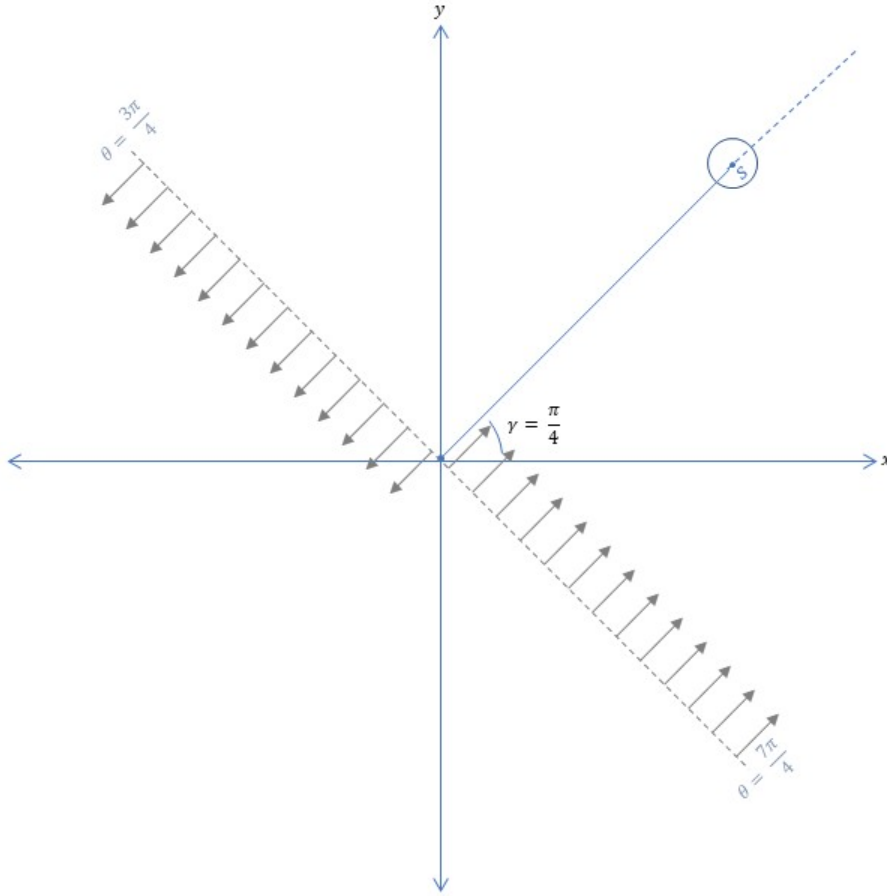
$$\begin{aligned} v_x = v_y &= \frac{c_x^*(t_1) - c_x^*(t_0)}{t_1 - t_0} \\ &= \frac{1}{\sqrt{2}(t_1 - t_0)} \left(\frac{z^*(t_1)}{\cos(\theta(t_1) - \frac{\pi}{4})} - \frac{z^*(t_0)}{\cos(\theta(t_0) - \frac{\pi}{4})} \right). \end{aligned}$$

We zullen deze formule toepassen in ons voorbeeld. Hierbij nemen we $t_1 = 1$ en $t_0 = 0$ en $\theta(0) = \theta(1) = 0$. We vinden $z^*(1) = 104$ en $z^*(0) = 32$. Deze waarden vullen we in en we krijgen:

$$\begin{aligned} v_x = v_y &= \frac{1}{\sqrt{2}(1-0)} \left(\frac{104}{\cos(0 - \frac{\pi}{4})} - \frac{32}{\cos(0 - \frac{\pi}{4})} \right) \\ &= \frac{1}{\sqrt{2}} \left(\frac{104}{\frac{1}{\sqrt{2}}} - \frac{32}{\frac{1}{\sqrt{2}}} \right) \\ &= 104 - 32 \\ &= 72 \text{ pixels / seconde} \end{aligned}$$

De dichtheid ρ van de schijf, hebben we in ons voorbeeld gezet op 1. Echter kan dit een willekeurige dichtheid zijn. We hadden voor de radontransformatie van de schijf gevonden:

$$q(\theta, z, t) = \begin{cases} 2\sqrt{r^2 - (z - s^* \cos(\theta - \gamma^*))^2} & , \text{ als } |z - s^* \cos(\theta - \gamma^*)| \leq r \\ 0 & , \text{ anders.} \end{cases}$$



Figuur 13: Ligging schijf ten op zichte van de projectie as voor $\theta \in \{\frac{3\pi}{4}, \frac{7\pi}{4}\}$

Geven we de schijf een willekeurige dichtheid ρ , dan verandert deze in:

$$q(\theta, z, t) = \begin{cases} 2\rho\sqrt{r^2 - (z - s^* \cos(\theta - \gamma^*))^2} & , \text{ als } |z - s^* \cdot \cos(\theta - \gamma^*)| \leq r \\ 0 & , \text{ anders} \end{cases}$$

met een maximum op $z = s^* \cos(\theta - \gamma^*)$. Dit geeft:

$$q^{max}(\theta, z) = 2\rho\sqrt{r^2} = 2\rho r.$$

De straal van de schijf is immers altijd positief. Als we de straal hebben gevonden, dan kunnen we de dichtheid van de schijf bepalen via de formule:

$$\rho = \frac{q^{max}(\theta, z)}{2r}.$$

Voor het berekenen van de straal, hebben we één projectie nodig waarbij we q^{max} halen uit de meting. Voor het bepalen van de dichtheid is ook één projectie nodig.

4.3.1 Meetschema (Stap 7)

We zullen nu een meetschema geven als samenvatting. Hierin is te zien hoeveel metingen minimaal nodig zijn om de gevraagde parameter terug te vinden.

Parameter	Aantal metingen
Straal: r	1
Dichtheid: ρ	1
Snelheid in de x - richting: v_x	2
Snelheid in de y - richting: v_y	2

4.4 Willekeurige lijn

We zullen nu langs een willekeurige lijn in het Euclidisch vlak bewegen waar we benieuwd zijn naar de parameters c_{x_0} , c_{y_0} , v_x en v_y . Hierbij zijn v_x en v_y als voorheen en (c_{x_0}, c_{y_0}) de coördinaten van het centrum van de disk op $t = t_0$.

Stelling

De parameters c_{x_0} , v_x , c_{y_0} en v_y zijn terug te halen uit de metingen als $\det(A) \neq 0$. Hierbij geldt:

$$A = \begin{pmatrix} \cos(\theta_1) & t_1 \cos(\theta_1) & \sin(\theta_1) & t_1 \sin(\theta_1) \\ \cos(\theta_2) & t_2 \cos(\theta_2) & \sin(\theta_2) & t_2 \sin(\theta_2) \\ \cos(\theta_3) & t_3 \cos(\theta_3) & \sin(\theta_3) & t_3 \sin(\theta_3) \\ \cos(\theta_4) & t_4 \cos(\theta_4) & \sin(\theta_4) & t_4 \sin(\theta_4) \end{pmatrix}$$

met t_i tijdstippen en θ_i hoeken.

Bewijs

Het enige wat we hebben zijn de metingen waarin we de projectie zien. Zoals eerder, zijn we alleen benieuwd naar de coördinaten van het middelpunt van de schijf op verschillende tijdstippen. In de projectie wordt deze gegeven daar waar de radontransformatie een maximum heeft, dus voor:

$$z^* = s \cos(\theta - \gamma)$$

met $s = \sqrt{c_x^2 + c_y^2}$ en $\gamma = \arctan(\frac{c_y}{c_x})$. Laten we z^* nu afhangen van de x - en y - coördinaten van het middelpunt van de schijf en van θ , dan krijgen we een gesloten uitdrukking voor de waarde van z daar waar een maximum is in de projectie. Dit geeft ons:

$$\hat{z}^*(x, y, \theta) = \sqrt{c_x^2 + c_y^2} \cos(\theta - \arctan(\frac{c_y}{c_x}))$$

Ter vergemakkelijking in berekeningen zullen we dit omschrijven. We gebruiken de identiteiten:

$$\cos(a - b) = \cos(a) \cos(b) + \sin(a) \sin(b)$$

$$\cos(\arctan(\frac{b}{a})) = \frac{a}{\sqrt{a^2+b^2}}$$

$$\sin(\arctan(\frac{b}{a})) = \frac{b}{\sqrt{a^2+b^2}}.$$

Er geldt:

$$\begin{aligned} \hat{z}^*(x, y, \theta) &= \sqrt{c_x^2 + c_y^2} \cdot \cos(\theta - \arctan(\frac{c_y}{c_x})) \\ &= \sqrt{c_x^2 + c_y^2} \cdot (\cos(\arctan(\frac{c_y}{c_x})) \cos \theta + \sin(\arctan(\frac{c_y}{c_x})) \sin \theta) \\ &= \sqrt{c_x^2 + c_y^2} \cdot (\frac{c_x}{\sqrt{c_x^2+c_y^2}} \cos \theta + \frac{c_y}{\sqrt{c_x^2+c_y^2}} \sin \theta) \\ &= c_x \cos(\theta) + c_y \sin(\theta) \end{aligned}$$

Dit is de uitdrukking waar we verder mee zullen werken. We herinneren dat we een dynamische schijf hebben, waarbij de schijf beweegt langs een lijn volgens de volgende formules:

$$c_x^*(t) = c_{x_0} + tv_x$$

$$c_y^*(t) = c_{y_0} + tv_y$$

Dit geeft de positie van het middelpunt van de schijf op verschillende tijdstippen. We laten θ nu een functie van t zijn, waarop \hat{z}^* een tijdsafhankelijke functie wordt:

$$\begin{aligned} \hat{z}^*(t) &= (c_{x_0} + t \cdot v_x) \cos(\theta(t)) + (c_{y_0} + t \cdot v_y) \sin(\theta(t)) \\ &= c_{x_0} \cos(\theta(t)) + t \cdot v_x \cos(\theta(t)) + c_{y_0} \sin(\theta(t)) + t \cdot v_y \sin(\theta(t)) \end{aligned}$$

We hebben 4 onbekenden, dus willen we vier vergelijkingen. Deze krijgen we door op vier verschillende tijdstippen $\hat{z}^*(t)$ uit de metingen te halen en deze in te vullen. Dit geeft:

$$\hat{z}^*(t_i) = c_{x_0} \cos(\theta_i) + t_i \cdot v_x \cos(\theta_i) + c_{y_0} \sin(\theta_i) + t_i \cdot v_y \sin(\theta_i) \text{ voor } i \in \{1, 2, 3, 4\}$$

Hierbij zetten we $\theta_i = \theta(t_i)$. Dit geeft een stelsel van vergelijkingen $A \vec{v} = \vec{\hat{z}^*}$ met $\vec{v} = (c_{x_0}, v_x, c_{y_0}, v_y)^\top$, $\vec{\hat{z}^*} = (\hat{z}^*(t_1), \hat{z}^*(t_2), \hat{z}^*(t_3), \hat{z}^*(t_4))^\top$ en

$$A = \begin{pmatrix} \cos(\theta_1) & t_1 \cos(\theta_1) & \sin(\theta_1) & t_1 \sin(\theta_1) \\ \cos(\theta_2) & t_2 \cos(\theta_2) & \sin(\theta_2) & t_2 \sin(\theta_2) \\ \cos(\theta_3) & t_3 \cos(\theta_3) & \sin(\theta_3) & t_3 \sin(\theta_3) \\ \cos(\theta_4) & t_4 \cos(\theta_4) & \sin(\theta_4) & t_4 \sin(\theta_4) \end{pmatrix}$$

Ons stelsel heeft een oplossing als A inverteerbaar is, oftewel als $\det(A) \neq 0$. We hebben:

$$\det(A) = \frac{1}{2}(-(t_1 - t_2)(t_3 - t_4) \cos(\theta_1 + \theta_2 - \theta_3 - \theta_4) + (t_1 - t_3)(t_2 - t_4) \cos(\theta_1 - \theta_2 + \theta_3 - \theta_4) - (t_1 - t_4)(t_2 - t_3) \cos(\theta_1 - \theta_2 - \theta_3 + \theta_4))$$

Voor een oplossing moet bovenstaande ongelijk aan 0 zijn. Dat is de voorwaarde die we stellen aan de matrix en daarmee dus aan de tijdstippen en hoeken. Die moeten zodanig zijn dat $\det(A) \neq 0$. \square

Stel dat beweging plaatsvindt over de lijn $y = x$. Dan geldt $c_{x_0} = c_{y_0}$ en $v_x = v_y$, en zijn we op zoek naar twee onbekenden in plaats van vier. Ons tijdafhankelijk schema is dan als volgt:

$$c_x^*(t) = c_y^*(t) = c_{x_0} + t \cdot v_x$$

Invullen geeft ons:

$$\begin{aligned} \hat{z}^*(t) &= 2(c_{x_0} + t \cdot v_x)(\cos(\theta(t)) + \sin(\theta(t))) \\ &= 2c_{x_0}(\cos(\theta(t)) + \sin(\theta(t))) + t \cdot 2v_x(\cos(\theta(t)) + \sin(\theta(t))) \end{aligned}$$

De vergelijkingen zien er nu als volgt uit:

$$\hat{z}^*(t_i) = 2c_{x_0}(\cos(\theta_i) + \sin(\theta_i)) + t \cdot 2v_x(\cos(\theta_i) + \sin(\theta_i)) \text{ voor } i \in \{1, 2\}$$

Onze matrix A is nu tweedimensionaal, met:

$$A = \begin{pmatrix} \cos(\theta_1) + \sin(\theta_1) & t_1(\cos(\theta_1) + \sin(\theta_1)) \\ \cos(\theta_2) + \sin(\theta_2) & t_2(\cos(\theta_2) + \sin(\theta_2)) \end{pmatrix}.$$

We merken op dat voor $\theta_i = \frac{3\pi}{4} + k\pi$, $k \in \mathbb{Z}$ geldt $\cos(\theta_i) = -\sin(\theta_i)$. Als $\theta_1 = \theta_2$ wordt iedere element in A nul. Als $\theta_i = \frac{3\pi}{4} + k\pi$ en $\theta_j \neq \frac{3\pi}{4} + k\pi$ voor $i \neq j$ wordt of de eerste rij of de tweede rij de nulrij. In het bijzonder betekent dit dat diens determinant nul is in alle drie de gevallen. Het stelsel van vergelijkingen is dan onoplosbaar en we kunnen niet de parameters c_{x_0} en v_x terug vinden uit de metingen volgens de stelling. Dit komt overeen met wat we eerder vonden in Hoofdstuk 4.3, namelijk de eis dat $\theta \notin \{\frac{3\pi}{4}, \frac{7\pi}{4}\}$.

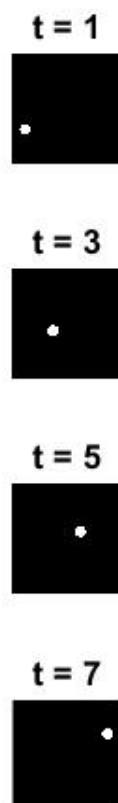
Stel nu dat $v_x \neq v_y$, maar wel gegeven. We genereren data via de ASTRA Toolbox en zullen A berekenen en bekijken of we v_x en v_y kunnen terugvinden middels de stelling.

We nemen $v_x = 80 \text{ pixels/seconde}$ en $v_y = 40 \text{ pixels/seconde}$. De schijf laten we beginnen op $(0, 160)$. Beweging vindt dus plaats langs de lijn $y = \frac{x}{2} + 160$ en $\vec{v} = (0, 80, 160, 40)^\top$.

We nemen $\vec{t} = (t_1, t_2, t_3, t_4)^\top = (1, 3, 5, 7)^\top$ met bijbehorende $\theta_i = \frac{(t_i - 1)\pi}{4}$. Dit geeft ons:

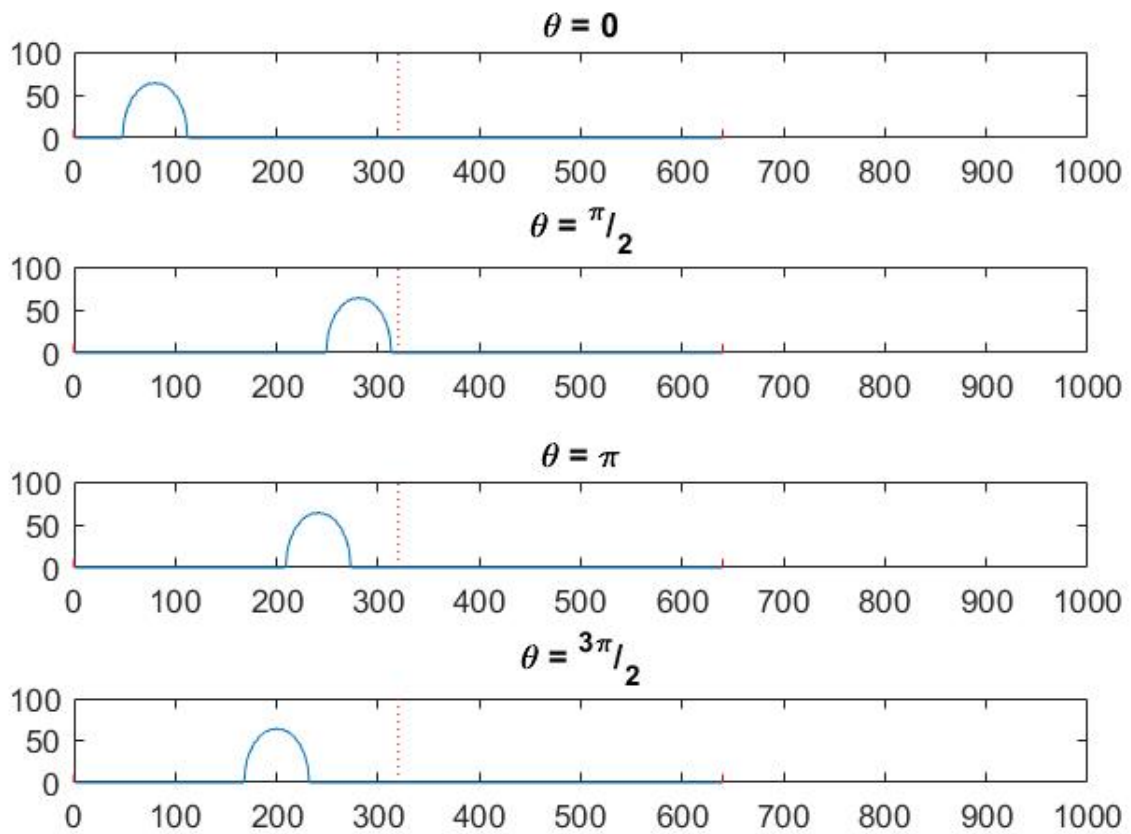
$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 \\ -1 & -5 & 0 & 0 \\ 0 & 0 & -1 & -7 \end{pmatrix}.$$

Deze matrix heeft determinant -16 . In figuur 14 zien we waar de schijf zich bevindt op verschillende tijdstippen. In figuur 15 zien we de projecties horend bij de verschillende hoeken θ_i . Uit dit figuur halen we de benodigde informatie om \vec{z}^* te bepalen. Dit doen we door het maximum te bepalen in iedere plot. We vinden $\vec{z}^* = (80, 281, -399, -440)^\top$. Voor $\theta = 0$ en $\theta = \frac{\pi}{2}$ kunnen we de plotjes gebruiken om z_1^* en z_2^* te achterhalen. Hier zijn dit de maximums van de plotjes. Voor $\theta = \pi$ en $\theta = \frac{3\pi}{2}$, dus voor $i \in \{3, 4\}$, geldt $z_i^* = z_i - 640$. Dit komt door hoe de ASTRA Toolbox werkt. Deze neemt altijd het centrum van het plaatje als midden van het domein van de projectie en roteert daar omheen, zie figuur 4. Dit heeft als gevolg dat we 640 van het berekende maximum af moeten trekken om z_i^* te bepalen. Voor het geval dat $\theta = \pi$, zie figuur 16.



Figuur 14: Positie van de cirkel in het vlak op verschillende tijdstippen.

Onze matrix A heeft een determinant ongelijk aan 0. De stelling vertelt ons dat we nu $\vec{v} = (c_{x_0}, v_x, c_{y_0}, v_y)^\top$ kunnen uitrekenen door de relatie:

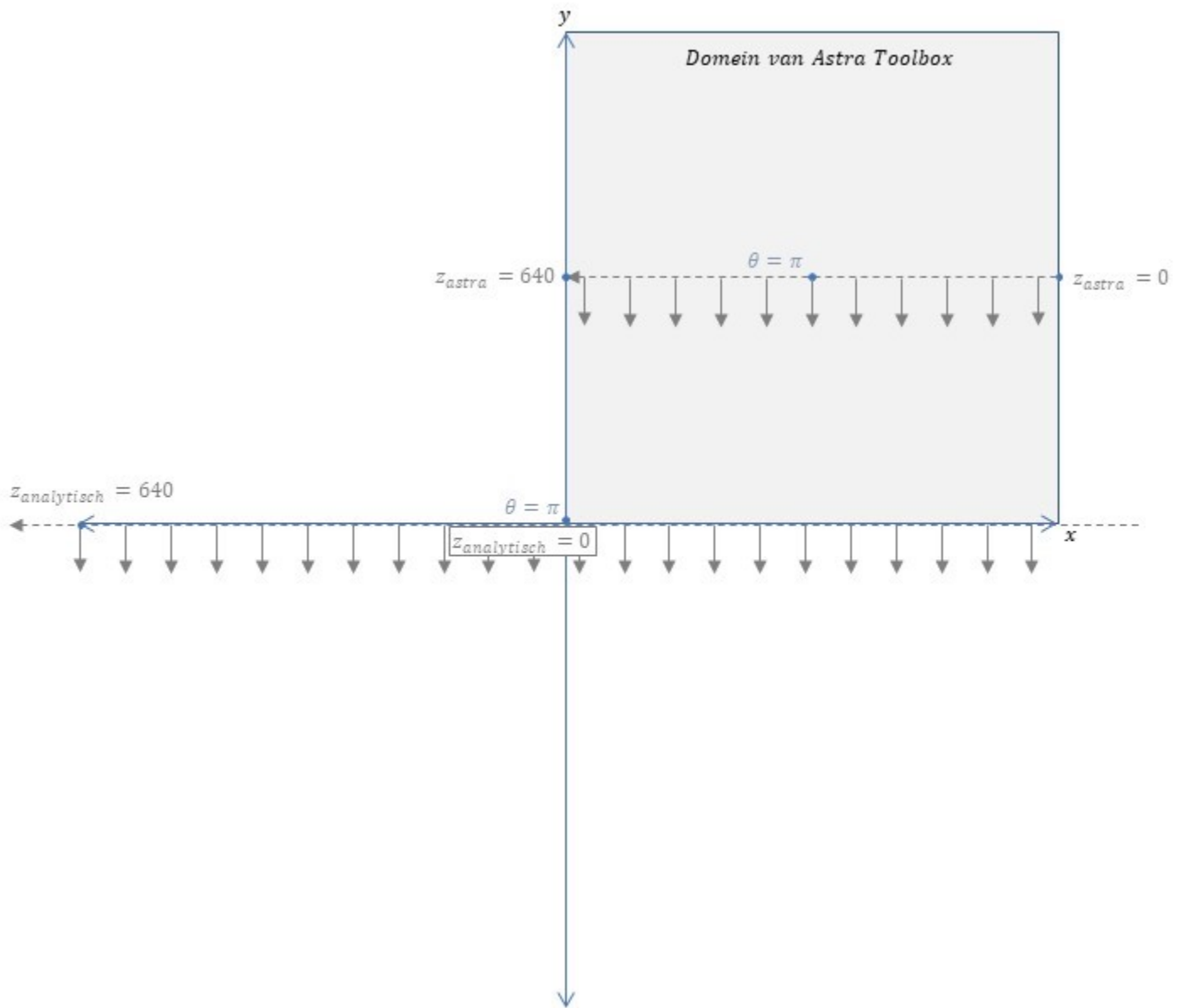


Figuur 15: Projectie (ASTRA Toolbox) van de cirkel door het vlak voor variërende θ op verschillende tijdstippen.

$$A\vec{v} = \vec{z}^*, \text{ oftewel}$$

$$\vec{v} = A^{-1}\vec{z}^*.$$

Dit geeft ons



Figuur 16: Positie van de z -as in de ASTRA Toolbox en analytisch voor $\theta = \pi$.

$$\begin{aligned}
 \vec{v} &= (c_{x_0}, v_x, c_{y_0}, v_y)^\top \\
 &= \frac{1}{4} \begin{pmatrix} 5 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \\ 0 & 7 & 0 & 3 \\ 0 & -1 & 0 & -1 \end{pmatrix} (80, 281, -399, -440)^\top \\
 &= \left(\frac{1}{4}, 79\frac{3}{4}, 161\frac{3}{4}, 39\frac{3}{4} \right)
 \end{aligned}$$

Dit zijn niet de exacte waarden, maar wel goede schattingen. We merken op dat voor $\vec{z}^* = (80, 280, -400, -440)^\top$ we wel de exacte waarde terug vinden. Bij het achterhalen van \vec{z}^* was er dus een fout in het berekenen van z_2^* en z_3^* .

We zullen nu enkele gevallen bekijken waarin hoeken en/of tijdstippen samenvallen en bekijken wat er gebeurt met $\det(A)$, met A gedefinieerd zoals in de stelling. In die gevallen zullen we iets vertellen over de oplosbaarheid van het systeem.

We beginnen met het geval waarin alle hoeken hetzelfde zijn. In dat geval geldt $\theta_1 = \theta_2 = \theta_3 = \theta_4$ en daarmee $\det(A) = 0$. Immers hebben we dan:

$$\begin{aligned} \det(A) &= \frac{1}{2}(-(t_1 - t_2)(t_3 - t_4) \cos(\theta_1 + \theta_1 - \theta_1 - \theta_1) + \\ &\quad (t_1 - t_3)(t_2 - t_4) \cos(\theta_1 - \theta_1 + \theta_1 - \theta_1) - \\ &\quad (t_1 - t_4)(t_2 - t_3) \cos(\theta_1 - \theta_1 - \theta_1 + \theta_1)) \\ &= \frac{1}{2}(-(t_1 - t_2)(t_3 - t_4) \cos(0) + (t_1 - t_3)(t_2 - t_4) \cos(0) - (t_1 - t_4)(t_2 - t_3) \cos(0)) \\ &= \frac{1}{2}(-(t_1 - t_2)(t_3 - t_4) + (t_1 - t_3)(t_2 - t_4) - (t_1 - t_4)(t_2 - t_3)) \\ &= 0 \end{aligned}$$

Het systeem is hiermee niet oplosbaar. Als we bedenken wat het betekent voor de metingen als $\theta_1 = \theta_2 = \theta_3 = \theta_4$, dan hebben we vier projectie's op (verschillende) tijdstippen onder dezelfde hoek genomen. Dit geeft niet genoeg informatie om de vier parameters terug te vinden.

Vervolgens bekijken we het geval waarin alle tijdstippen hetzelfde zijn, dus $t_1 = t_2 = t_3 = t_4$. Hier geldt wederom $\det(A) = 0$, want:

$$\begin{aligned} \det(A) &= \frac{1}{2}(-(t_1 - t_1)(t_1 - t_1) \cos(\theta_1 + \theta_2 - \theta_3 - \theta_4) + \\ &\quad (t_1 - t_1)(t_1 - t_1) \cos(\theta_1 - \theta_2 + \theta_3 - \theta_4) - \\ &\quad (t_1 - t_1)(t_1 - t_1) \cos(\theta_1 - \theta_2 - \theta_3 + \theta_4)) \\ &= \frac{1}{2}(-(0)(0) \cos(\theta_1 + \theta_2 - \theta_3 - \theta_4) + (0)(0) \cos(\theta_1 - \theta_2 + \theta_3 - \theta_4) - \\ &\quad (0)(0) \cos(\theta_1 - \theta_2 - \theta_3 + \theta_4)) \\ &= 0 \end{aligned}$$

Dit systeem is ook niet oplosbaar. We maken op dezelfde tijdstip al onze metingen. Hiermee zullen we nooit de snelheid kunnen vinden in of de x - of de y - richting. Hiervoor hebben we meer tijdstippen nodig.

Dit waren triviale gevallen. We zullen nu een minder triviale geval bekijken, het geval waarin $t_1 = t_2$, $t_3 = t_4$, $\theta_1 = \theta_3$ en $\theta_2 = \theta_4$. We hebben dus twee tijdstippen, met op ieder tijdstip twee hoeken waarop we metingen maken. Onze eerste indruk is dat het systeem dan ook onoplosbaar is en we geen parameters terug kunnen halen. Berekenen we de determinant

van A , dan vinden we:

$$\begin{aligned}
 \det(A) &= \frac{1}{2}(-(t_1 - t_1)(t_3 - t_3) \cos(\theta_1 + \theta_2 - \theta_1 - \theta_2) + \\
 &\quad (t_1 - t_3)(t_1 - t_3) \cos(\theta_1 - \theta_2 + \theta_1 - \theta_2) - \\
 &\quad (t_1 - t_3)(t_1 - t_3) \cos(\theta_1 - \theta_2 - \theta_1 + \theta_2)) \\
 &= \frac{1}{2}(-(0)(0) \cos(0) + (t_1 - t_3)^2 \cos(2\theta_1 - 2\theta_2) - (t_1 - t_3)^2 \cos(0)) \\
 &= \frac{1}{2}((t_1 - t_3)^2 \cos(2\theta_1 - 2\theta_2) - (t_1 - t_3)^2)
 \end{aligned}$$

Dit is gelijk aan nul dan en slechts dan als $t_1 = t_3$ of $\theta_2 = \theta_1 - k\pi$, $k \in \mathbb{Z}$. In alle andere gevallen geldt $\det(A) \neq 0$ en is het systeem oplosbaar. In die gevallen kunnen we de parameters terug vinden. We kunnen als voorbeeld het eerder besproken geval bekijken, waarbij we langs de lijn $y = x$ bewegen en als startpunt de oorsprong hebben. Hier zagen we een oplossing als $\theta_i \notin \{\frac{3\pi}{4}, \frac{7\pi}{4}\}$.

5 Rotatie van eencelligen

Protoplasten zijn cellen die niet of gedeeltelijk worden omgeven door een celwand. Bij protoplasten van *Avena Sativa* (Haver) vindt rotatie plaats in een alternerende elektrische veld, als minstens twee cellen dicht op elkaar zijn gelegen^[5]. Het roteren van een cel kunnen we simuleren aan de hand van een roterende ellips. Op de roterende ellips zullen we tomografie toepassen om enkele parameters terug te vinden. Hierbij zullen we ons stappenplan aanhouden. We beginnen met het analytische gedeelte en zullen hierna numeriek parameters proberen terug te vinden.

5.1 Analytisch

Stap 1

We beginnen met het maken van een ellips met willekeurige middelpunt (c_x, c_y) in het vlak. Dit doen we aan de hand van de volgende functie:

$$f_{A,B,c_x,c_y}(x,y) = \begin{cases} 1 & , \text{ als } \frac{(x-c_x)^2}{A^2} + \frac{(y-c_y)^2}{B^2} \leq 1 \\ 0 & , \text{ anders} \end{cases}$$

Hierbij zijn A en B de parameters die de vorm van de ellips meegeven. Als $A = B$, dan hebben we een cirkel.

Stap 2

Voor f vinden we een projectie P onder hoek θ [6]:

$$P(\theta, z) = \begin{cases} \frac{2AB}{r(\theta)^2} \sqrt{r(\theta)^2 - (z - s \cos(\theta - \gamma))^2} & , \text{ als } |z - s \cos(\theta - \gamma)| \leq r(\theta) \\ 0 & , \text{ anders} \end{cases}$$

met $s = \sqrt{c_x^2 + c_y^2}$, $\gamma = \arctan \frac{c_y}{c_x}$ en $r(\theta)^2 = A^2 \cos^2(\theta) + B^2 \sin^2(\theta)$

We merken op dat dit een generalisatie is van de projectie van de schijf met dichtheid $\rho = 1$. Voor de schijf geldt $A = B = r$. Voor het gemak houden we als middelpunt $(c_x, c_y) = (0, 0)$ aan. Dit versimpelt de projectie tot:

$$P(\theta, z) = \begin{cases} \frac{2AB}{r(\theta)^2} \sqrt{r(\theta)^2 - z^2} & , \text{ als } |z| \leq r(\theta) \\ 0 & , \text{ anders} \end{cases}$$

met $r(\theta) = A^2 \cos^2(\theta) + B^2 \sin^2(\theta)$.

Stap 3

We hebben een statische ellips gemaakt en bijbehorende projectie gevonden. Nu zullen we de ellips laten roteren. Om de ellips te laten roteren introduceren we een hoek $\omega(t)$. Dit is de hoek waarmee is geroteerd op een bepaald tijdstip. In de volgende stap wordt hiermee de dynamische ellips gedefinieerd. De ellips zal een bepaalde rotatiesnelheid $\omega'(t)$ aanhouden bij het roteren. Deze zullen we later proberen terug te vinden.

Stap 4

De roterende ellips, met als middelpunt de oorsprong, beschrijven we nu met de volgende functie:

$$g_{A,B,t}(x, y) = \begin{cases} 1 & , \text{ als } \frac{(x \cos(\omega(t)) + y \sin(\omega(t)))^2}{A^2} + \frac{(x \sin(\omega(t)) - y \cos(\omega(t)))^2}{B^2} \leq 1 \\ 0 & , \text{ anders} \end{cases}$$

Hierbij is $\omega(t)$ de hoek waarover is geroteerd op een bepaald tijdstip.

Stap 5

De projectie wat hierbij hoort is als volgt:

$$q(\theta, z, t) = \begin{cases} \frac{2AB}{r(t)^2} \sqrt{r(t)^2 - z^2} & , \text{ als } |z| \leq r(t) \\ 0 & , \text{ anders} \end{cases}$$

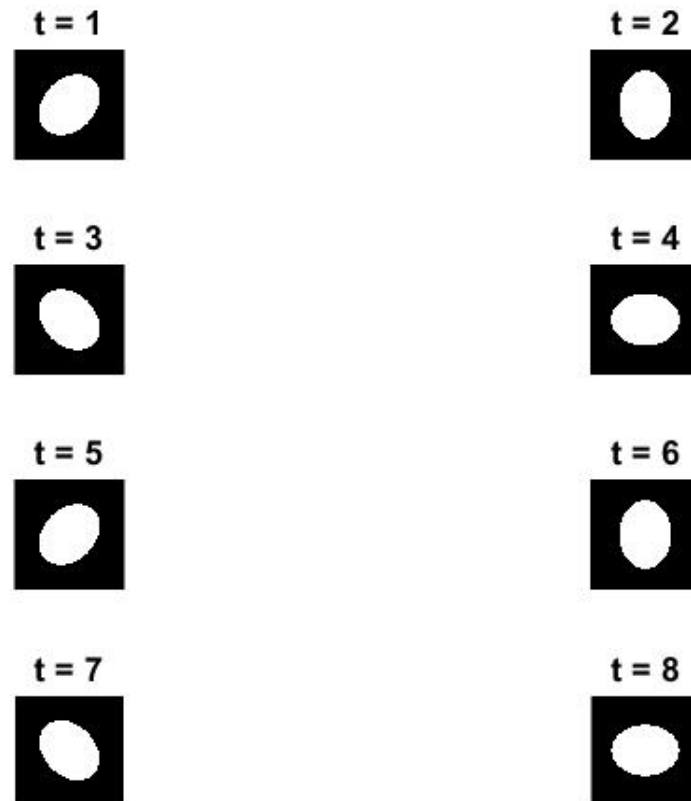
met $r(t)^2 = (B^2 \cos^2(\omega(t)) + A^2 \sin^2(\omega(t))) \cos^2(\theta(t)) + (B^2 \sin^2(\omega(t)) - A^2 \cos^2(\omega(t))) \sin^2(\theta(t))$. Hierbij zijn $\omega(t)$ en $\theta(t)$ respectievelijk de hoek van rotatie en hoek waarover de projectie gemaakt wordt op een bepaalde tijdstip. De projectie vinden we door de projectie in stap 2 en de voorwaarde in stap 4 voor het maken van de roterende ellips te gebruiken.

5.2 Numeriek

We creëren een pixelafbeelding met maten 641 x 641. De ellips plaatsen we in het midden van de afbeelding. Deze geven we de parameterwaarden $A = 200$ en $B = 150$ mee. We gaan ervan uit dat we niet weten hoe de ellips is geroteerd in zijn beginpositie. Verder vindt rotatie plaats over 2π gedurende 8 seconden. We hebben dus een rotatiesnelheid $\omega' = \frac{\pi i}{4} \text{ rad} / \text{sec}$.

De projecties zullen we numeriek vinden via de ASTRA Toolbox. Deze zullen we gebruiken om de parameters A, B en ω' terug te vinden. Ook zullen we deze gebruiken om onze analytische functie voor de projectie te controleren. Omdat de ellips roteert om een vast punt, nemen we één hoek θ . Hiervoor nemen we altijd $\theta = 0$. Hierdoor zullen we projecties van de ellips zien voor verschillende hoeken ω . Voor de hoek van rotatie $\omega(t)$ houden we de functie $\omega(t) = \frac{\pi}{4}t$ aan.

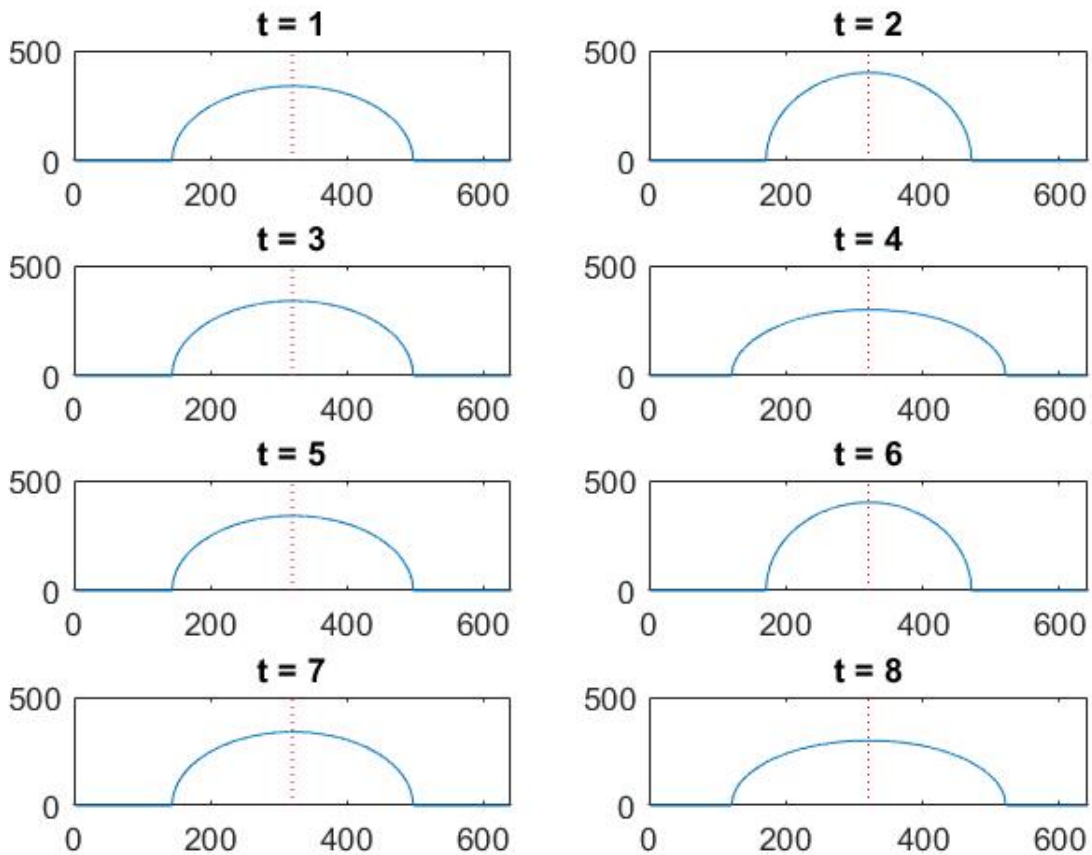
In figuur 17 zien we de ligging van de ellips op verschillende tijdstippen. De bijbehorende projecties zien we in figuur 18.



Figuur 17: Ligging van de ellips op de tijdstippen $t = 1, \dots, 8$.

In figuur 18 is het midden van het domein voor de projectie weergegeven door een stippelijntje. Voor het maken van deze figuren hebben een programma geschreven wat overeen komt met de programma gebruikt in hoofdstuk 6. Zie de bijlage voor de volledige code. Eerst maken we een afbeelding van 641×641 pixels en beschrijven we de tijdsinterval, rotatie en de hoeken waarover we projecties maken. Hierna simuleren we een roterende schijf, waarvan we op verschillende tijdstippen projecties maken.

In figuur 19 zien we een plot van de projectie die we analytisch hebben gevonden. Als we deze vergelijken met de projectie gevonden via de ASTRA Toolbox, dan zien we dat de twee hetzelfde zijn.

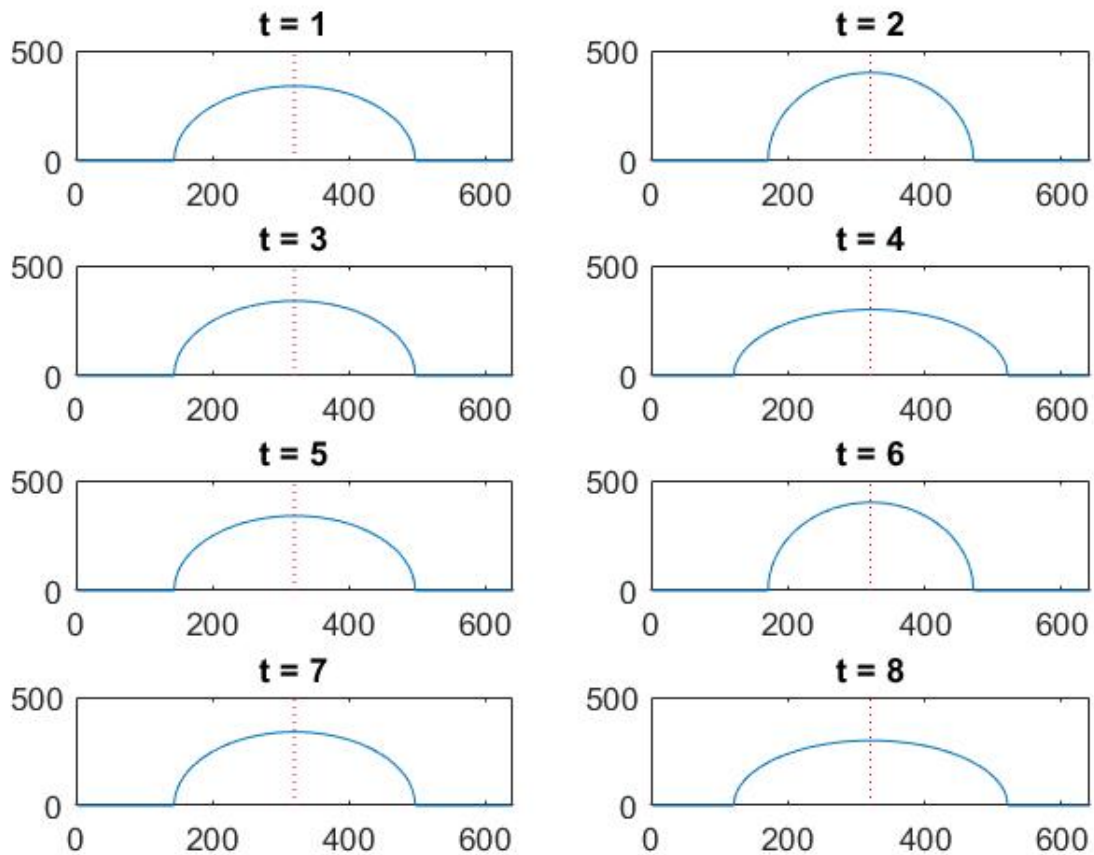


Figuur 18: Projectie (ASTRA Toolbox) van de ellips op de tijdstippen $t = 1, \dots, 8$.

5.3 Terugvinden van Parameters (Stap 6)

We zullen nu de parameters A , B en ω' terugvinden aan de hand van de projecties. Hierbij bekijken we ook hoeveel projecties onder hoeveel hoeken we nodig hebben om de parameters te vinden.

We beginnen met de parameters A en B die de vorm van de ellips beschrijven. Omdat we ervan uit gaan dat de rotatie van de ellips op $t = 0$ onbekend is, kunnen we geen tijdstippen kiezen. We zullen kijken we naar de breedte van de projecties. De maximale en minimale breedte zullen ons de parameters A en B geven. De projecties hebben immers waarde 0 wanneer de stralen niet door de ellips gaan en een positieve waarde wanneer de stralen wel door de ellips gaan. Als we een maximale breedte vinden, dan zal dat de grootste parameter van de ellips geven. Als we een minimale breedte vinden, dan zal dat de kleinste parameter van de ellips geven. Het maakt niet uit welke parameter we A en welke we B noemen. Zonder verlies van algemeenheid noemen we de kleinste parameter A en de grootste



Figuur 19: Projectie (Analytisch) van de ellips op de tijdstippen $t = 1, \dots, 8$.

B. We vinden A en B via de formules:

$$A = \frac{\text{minimale breedte projectie}}{2}$$

$$B = \frac{\text{maximale breedte projectie}}{2}$$

Dit geeft ons de waarden $A = 150.5$ en $B = 200.5$. De oorspronkelijke waarden waren $A = 200$ en $B = 150$. Op wisseling van A en B na hebben we een redelijk nauwkeurige schatting. De projectie is een halve ellips wat varieert in vorm. Als we de breedte van de halve ellips bijhouden met een teller en we vinden een waarde waarna de breedte van de ellips daalt na eerst gestegen te hebben, dan hebben we een maximum gevonden. Als we een waarde vinden waarna de breedte van de ellips stijgt na eerst gedaald te hebben, dan hebben we een minimum gevonden. We maken dus projecties tot we beide gevallen zijn tegengekomen.

Voor de rotatiesnelheid houden we bij hoe de projectie eruitziet. Omdat de projectie varieert in vorm, heeft het geen vaste maximum. Het heeft echter wel een echte maximale waarde. Dat is wanneer de langste as van de ellips parallel loopt aan de stralen waarlangs we de projectie maken, dus langs de y -as. We noemen deze maximale waarde het maximum. We houden tijdstippen bij waarop de projectie een maximum heeft. Wanneer het maximum twee keer wordt aangenomen, heeft de ellips een rotatie gemaakt van π radialen. We meten de twee tijdstippen waarop de projectie een maximum heeft. Dit geeft ons een maximum van 401 op $t = 2$ en op $t = 6$. We vinden een rotatiesnelheid:

$$\omega' = \frac{\pi}{6-2} = \frac{\pi}{4} \text{ rad/sec.}$$

Het aantal projecties nodig voor het vinden van de rotatiesnelheid is ook niet expliciet. We maken projecties totdat we twee keer het maximum van de projecties zijn tegengekomen. Dat aantal kan variëren. De rotatie van de ellips in de begintoestand heeft hier een invloed op.

5.3.1 Meetschema

We zullen nu een meetschema geven als samenvatting. Hierin is te zien hoeveel metingen minimaal nodig zijn om de gevraagde parameter terug te vinden.

Parameter	Aantal metingen
Vormgever: A	x aantal totdat we een waarde hebben gevonden waarna de breedte van de ellips daalt na eerst gestegen te hebben
Vormgever: B	x aantal totdat we een waarde hebben gevonden waarna de breedte van de ellips stijgt na eerst gedaald te hebben
Rotatiesnelheid: ω'	x aantal totdat we twee keer het maximum van de projecties zijn tegengekomen

6 Conclusie

In ons onderzoek zijn we begonnen met het formuleren van de Fourierslice Stelling en bekijken wat van de stelling centraal zou staan in ons onderzoek. Dit was de radontransformatie. De radontransformatie van een object voor een hoek θ beschrijft de projectie van een object over de hoek θ . Een projectie krijgen we als we stralen vanuit een sensor door het object sturen. Deze stralen nemen informatie mee die aan de andere kant van het object door een scanner worden opgemeten. We hebben de ASTRA Toolbox in MATLAB gebruikt om projecties te maken. We zagen dat in het geval van een disk met een bepaalde dichtheid, de projectie een halve ellips is. De projecties hebben we gebruikt om relevante eigenschappen van een object terug te vinden. Dit waren eigenschappen waarmee we het object zouden kunnen classificeren.

We hebben dynamische systemen onderzocht. Om met behulp van tomografie hier parameters van objecten uit te krijgen, hebben we een stappenplan gemaakt. Met deze stappenplan kunnen we met een minimaal aantal projecties, de gevraagde parameters terugvinden. In het bijzonder hebben we twee systemen bekeken.

We begonnen met een disk die langs een lijn beweegt in het Euclidisch vlak. Het bleek dat we de parameters voor de straal en dichtheid terug konden vinden met 1 projectie. Hier hadden we wel de analytisch uitdrukking nodig voor het terugvinden van de dichtheid. Voor de positie (c_x, c_y) van het centrum en snelheden v_x en v_y in respectievelijk de x - en y -richting hebben we een stelling gevonden. Helaas gaf de stelling geen garantie voor het vinden van de exacte waarden van de parameters. Het geeft wel een goede schatting. In al de gevallen konden we een expliciet aantal geven voor het minimaal aantal projecties nodig voor het vinden van de parameters.

Hierna hebben we rotatie van eencelligen bekeken. Hiertoe hebben we een ellips laten roteren om de oorsprong. We wilden de parameters die de ellips vorm geven en de rotatiesnelheid terugvinden. Met de hulp van ons stappenplan hebben we de gevraagde parameters terug weten te vinden. Voor de vormparameters vonden we erg goede schattingen en voor de rotatiesnelheid vonden we de exacte waarde.

We concluderen dat ons stappenplan de gevraagde parameters terugvindt met een minimaal aantal benodigde projecties. De gevonden meetwaarden gaven echter niet altijd de exacte parameterwaarden. Dit kan komen door meetfouten. Ook speelt de grootte van de pixelafbeelding een rol. Bij een grotere pixelafbeelding zullen de schattingen veel beter overeenkomen met de exacte parameterwaarden.

7 Bijlage: Code

Code gebruikt in hoofdstuk 4 en 6

De figuren 5, 6, 8, 9, 10, 11 en 12 zijn gemaakt met de volgende code:

```
clear all; close all;
imagesizeX = 640;
imagesizeY = 640;
radius = imagesizeX/20;
grad = 1;
circle = zeros(imagesizeX, imagesizeY);
te = 8;
dy = imagesizeY-radius:-1:radius;
dx = -grad*dy+imagesizeX;
t = 0:te/(length(dx)-1):te;
theta = 0:pi/4:2*pi;
%theta = zeros(9,1)+0*pi/4;
teller = 1;
for k = 1:length(t)
    for i = 1:imagesizeY
        for j = 1:imagesizeX
            if (i-dy(k))^2+(j-dx(k))^2 <= radius^2
                circle(i,j) = 1;
            else
                circle(i,j) = 0;
            end
        end
    end
end

if (t(k) == 1 || t(k) == 2 || t(k) == 3 || t(k) == 4 || t(k) == 5 ||
t(k) == 6 || t(k) == 7 || t(k) == 8)
    cx = teller*(imagesizeX-imagesizeX/10)/8+imagesizeX/20;
    cy = teller*(imagesizeY-imagesizeY/10)/8+imagesizeY/20;
    j = atan(cy/cx);
    s = sqrt(cx^2+cy^2);
    if (theta(teller) == 3*pi/4 || theta(teller) == 7*pi/4)
        detcount = floor(sqrt(imagesizeX^2+imagesizeY^2));
    else
        detcount = floor(abs(sqrt(imagesizeX^2+imagesizeY^2)*
        cos(theta(teller)-j)));
    end
    vol_geom = astra_create_vol_geom(imagesizeX, imagesizeY);
    proj_geom = astra_create_proj_geom('parallel', 1.0,
    detcount, theta(teller));
```



```

sinogramx = 1:detcount;
[sinogram_id, sinogram] = astra_create_sino_gpu(circle,
proj_geom, vol_geom);
if (theta(teller) < 3*pi/4 || (7*pi/4 < theta(teller) &&
theta(teller) <= 2*pi))
    p = 2*sqrt(radius^2-(sinogramx-s*cos(theta(teller)-j)).^2);
end
if (3*pi/4 < theta(teller) && theta(teller) < 7*pi/4)
    p = 2*sqrt(radius^2-(sinogramx-s*cos(theta(teller)-j)-
detcount).^2);
end
if (theta(teller) == 7*pi/4 || theta(teller) == 3*pi/4)
    p = 2*sqrt(radius^2-(sinogramx-s*cos(theta(teller)-j)-
0.5*sqrt(imagesizeX^2+imagesizeY^2)).^2);
end
figure(1)
hold on
subplot(4,2,teller)
imshow(circle)
if (t(k)==1)
    title('t = 1');
end
if (t(k)==2)
    title('t = 2');
end
if (t(k)==3)
    title('t = 3');
end
if (t(k)==4)
    title('t = 4');
end
if (t(k)==5)
    title('t = 5');
end
if (t(k)==6)
    title('t = 6');
end
if (t(k)==7)
    title('t = 7');
end
if (t(k)==8)
    title('t = 8');
end
hold off
figure(2)

```

```

hold on
subplot(4,2,teller)
plot(sinogramx,sinogram,[detcount/2 detcount/2],[0 100],':r',[0 0],[0 10],':r',[detcount detcount],[0 10],':r')
axis([0 1000 0 100])
if (t(k)==1)
    title('\theta = 0');
end
if (t(k)==2)
    title('\theta = ^{\pi}/_{4}');
end
if (t(k)==3)
    title('\theta = ^{2\pi}/_{4}');
end
if (t(k)==4)
    title('\theta = ^{3\pi}/_{4}');
end
if (t(k)==5)
    title('\theta = \pi');
end
if (t(k)==6)
    title('\theta = ^{5\pi}/_{4}');
end
if (t(k)==7)
    title('\theta = ^{6\pi}/_{4}');
end
if (t(k)==8)
    title('\theta = ^{7\pi}/_{4}');
end
hold off
figure(3)
hold on
subplot(4,2,teller)
plot(z,p,[detcount/2 detcount/2],[0 100],':r',[0 0],[0 10],':r',[detcount detcount],[0 10],':r')
axis([0 1000 0 100])
if (t(k)==1)
    title('\theta = 0');
end
if (t(k)==2)
    title('\theta = ^{\pi}/_{4}');
end
if (t(k)==3)
    title('\theta = ^{2\pi}/_{4}');
end
end

```

```

    if (t(k)==4)
        title ('\ theta = ^{3\pi}/_{4} ');
    end
    if (t(k)==5)
        title ('\ theta = \pi ');
    end
    if (t(k)==6)
        title ('\ theta = ^{5\pi}/_{4} ');
    end
    if (t(k)==7)
        title ('\ theta = ^{6\pi}/_{4} ');
    end
    if (t(k)==8)
        title ('\ theta = ^{7\pi}/_{4} ');
    end
    hold off
    astra_mex_data2d('delete', sinogram_id);
    teller = teller+1;
end%if
end%for

```

In hoofdstuk 6.4 testen we onze stelling. De figuren 14 en 15 die hiervoor zijn gebruikt zijn gemaakt met de volgende code:

```

clear all; close all;
imagesizeX = 640;
imagesizeY = 640;
radius = imagesizeX/20;
grad = 1;
circle = zeros(imagesizeX, imagesizeY);
te = 8;
dy = imagesizeY - 160: -0.5: 160;
dx = -2*grad*dy+imagesizeX+320;
t = 0:te/(length(dx)-1):te;
theta = 0:pi/4:2*pi;
centre = length(circle)/2;
teller = 1;
num = 1;
d = zeros(4,1);
for k = 1:length(t)
    for i = 1:imagesizeY
        for j = 1:imagesizeX
            if (i-dy(k))^2+(j-dx(k))^2 <= radius^2

```

```

        circle(i,j)= 1;
    else
        circle(i,j)= 0;
    end
end
end
if (t(k) == 1 || t(k) == 3 || t(k) == 5 || t(k) == 7)
    detcount = imagesizeX;
    vol_geom = astra_create_vol_geom(imagesizeX, imagesizeY);
    proj_geom = astra_create_proj_geom('parallel', 1.0,
    detcount, theta(teller));
    sinogramx = 1:detcount;
    [sinogram_id, sinogram] = astra_create_sino_gpu(circle,
    proj_geom, vol_geom);
    for j = 1:detcount
        max(sinogram)
        if sinogram(j) == max(sinogram)
            d(num) = j;
        end
    end
    figure(1)
    hold on
    subplot(4,1,num)
    imshow(circle)
    if (t(k)==1)
        title('t = 1');
    end
    if (t(k)==3)
        title('t = 3');
    end
    if (t(k)==5)
        title('t = 5');
    end
    if (t(k)==7)
        title('t = 7');
    end
    hold off
    figure(2)
    hold on
    subplot(4,1,num)
    plot(sinogramx, sinogram, [detcount/2 detcount/2], [0 100], 'r',
    [0 0], [0 10], 'r', [detcount detcount], [0 10], 'r')
    axis([0 1000 0 100])
    if (t(k)==1)
        title('\theta = 0');
    end
end
end

```

```

end
if (t(k)==3)
    title('\theta = ^{\pi}/_{2}');
end
if (t(k)==5)
    title('\theta = \pi');
end
if (t(k)==7)
    title('\theta = ^{3\pi}/_{2}');
end
hold off
astra_mex_data2d('delete', sinogram_id);
teller = teller+2;
num = num+1;
end%if
end%for
d

```

Code gebruikt in hoofdstuk 7

De figuren 17, 18 en 19 zijn gemaakt met de volgende code:

```

clear all; close all;
imagesizeX = 641;
imagesizeY = 641;
cx = ceil(imagesizeX/2);
cy = ceil(imagesizeY/2);
A = 150;
B = 200;
circle = zeros(imagesizeX, imagesizeY);
te = 8;
t = 0:0.05:te;
w = 0:2*pi/(length(t)-1):2*pi;
theta = 0;
teller = 1;
mini = 1000;
maxi = 0;
count = 0;
num = 1;
time = zeros(1,2);
rotmax = zeros(1,length(t));
schakel = 0;
for k = 1:length(t)
    for i = 1:imagesizeY

```

```

    for j = 1:imagesizeX
        if (((j-cx)*sin(w(k))+(i-cy)*cos(w(k)))/A)^2+
            (((j-cx)*cos(w(k))-(i-cy)*sin(w(k)))/B)^2 <= 1
            circle(i,j)= 1;
        else
            circle(i,j)= 0;
        end
    end
end
vol_geom = astra_create_vol_geom(imagesizeX, imagesizeY);
proj_geom = astra_create_proj_geom('parallel', 1.0,
imagesizeX, theta);
sinogramx = 1:imagesizeX;
[sinogram_id, sinogram] = astra_create_sino_gpu(circle,
proj_geom, vol_geom);
rkwad = (B^2*cos(w(k))^2+A^2*sin(w(k))^2)*cos(theta)^2+
(B^2*sin(w(k))^2-A^2*cos(w(k))^2)*sin(theta)^2;
p = ((2*A*B)/rkwad)*sqrt(rkwad-(sinogramx-cx).^2);
for j = 1:imagesizeX
    if sinogram(j) ~= 0
        count = count + 1;
    end
end
if count > maxi
    maxi = count;
end
if count < mini && count ~= 0
    mini = count;
end
rotmax(num) = max(sinogram);
num = num+1;
count = 0;
if (t(k) == 1 || t(k) == 2 || t(k) == 3 || t(k) == 4 || t(k) == 5 ||
t(k) == 6 || t(k) == 7 || t(k) == 8)
    figure(1)
    hold on
    subplot(4,2,teller)
    imshow(circle)
    if (t(k)==1)
        title('t = 1');
    end
    if (t(k)==2)
        title('t = 2');
    end
    if (t(k)==3)

```

```

        title('t = 3');
end
if (t(k)==4)
    title('t = 4');
end
if (t(k)==5)
    title('t = 5');
end
if (t(k)==6)
    title('t = 6');
end
if (t(k)==7)
    title('t = 7');
end
if (t(k)==8)
    title('t = 8');
end
hold off
figure(2)
hold on
subplot(4,2,teller)
plot(sinogramx,sinogram,[ceil(imagesizeX)/2 ceil(imagesizeX)/2],
[0 500],':r',[0 0],[0 10],':r',[imagesizeX imagesizeX],[0 10],':r')
axis([0 imagesizeX 0 500])
if (t(k)==1)
    title('t = 1');
end
if (t(k)==2)
    title('t = 2');
end
if (t(k)==3)
    title('t = 3');
end
if (t(k)==4)
    title('t = 4');
end
if (t(k)==5)
    title('t = 5');
end
if (t(k)==6)
    title('t = 6');
end
if (t(k)==7)
    title('t = 7');
end
end

```

```

    if (t(k)==8)
        title('t = 8');
    end
    hold off
    figure(3)
    hold on
    subplot(4,2,teller)
    plot(sinogramx,p,[ceil(imagesizeX)/2 ceil(imagesizeX)/2],[0 500],
        ':r',[0 0],[0 10], 'r',[imagesizeX imagesizeX],[0 10], 'r')
    axis([0 imagesizeX 0 500])
    if (t(k)==1)
        title('t = 1');
    end
    if (t(k)==2)
        title('t = 2');
    end
    if (t(k)==3)
        title('t = 3');
    end
    if (t(k)==4)
        title('t = 4');
    end
    if (t(k)==5)
        title('t = 5');
    end
    if (t(k)==6)
        title('t = 6');
    end
    if (t(k)==7)
        title('t = 7');
    end
    if (t(k)==8)
        title('t = 8');
    end
    hold off
    teller = teller+1;
end%end
astra_mex_data2d('delete', sinogram_id);
end%end
for i = 1:length(rotmax)
    if rotmax(i) == max(rotmax) && schakel == 0
        time(1) = t(i);
        schakel = 1;
    end
    if rotmax(i) == max(rotmax) && schakel == 1

```



```
        time(2) = t(i)
    end
end
maxi
mini
time
```

Referenties

- [1] Timothy G. Feeman. (2015) *The Mathematics of Medical Imaging*, Editie: 2. Springer International Publishing. [Bladzijde 15].
- [2] W. van Aarle, W. J. Palenstijn, J. Cant, E. Janssens, F. Bleichrodt, A. Dabravolski, J. De Beenhouwer, K. J. Batenburg en J. Sijbers. (2016) *Fast and Flexible X-ray Tomography Using the ASTRA Toolbox*. Optics Express, 24(22), 25129-25147. <http://dx.doi.org/10.1364/OE.24.025129>
- [3] W. van Aarle, W. J. Palenstijn, J. De Beenhouwer, T. Altantzis, S. Bals, K. J. Batenburg en J. Sijbers. (2015) *The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography*. Ultramicroscopy, 157, 35-47. <http://dx.doi.org/10.1016/j.ultramic.2015.05.002>
- [4] W. J. Palenstijn, K. J. Batenburg en J. Sijbers. (2011) *Performance improvements for iterative electron tomography reconstruction using graphics processing units (GPUs)*. Journal of Structural Biology, vol. 176, editie 2, blz. 250-253, <http://dx.doi.org/10.1016/j.jsb.2011.07.017>
- [5] C. Holzapfel, J. Vienken en U. J. Zimmermann (1982) *Membrain Biol.* 67:13. <https://doi.org/10.1007/BF01868644>
- [6] A. C. Kak en Malcolm Slaney. (2001) *Principles of Computerized Tomographic Imaging*. Society of Industrial and Applied Mathematics. [Bladzijde 53-56]