

Kompendium

Jonas Persson og Sander Ruud

# Videoanalyse i fysikkundervisningen

Trondheim, juli 2021

**NTNU**  
Norges teknisk-naturvitenskapelige universitet  
Fakultet for naturvitenskap  
Institutt for fysikk

Videoanalyse i fysikkundervisningen, v3.0  
ISBN 978-82-8054-205-2  
Juli 2021  
Institutt for fysikk  
NTNU, 7491 Trondheim

# Videoanalyse i fysikkundervisningen

Jonas Persson og Sander Ruud  
Institutt for fysikk  
Norges teknisk-naturvitenskapelige universitet

Trondheim, Juli 2021



# Forord

Videoanalys och dess nytta i fysikundervisning är väl etablerad. Dagens elever är väl förtroliga med videoformatet och har idag möjlighet att själva producera video med hög kvalitet. I en video finns det förutom en stor mängd information om rummet även tidsinformation, något som utnyttjas inom fysiken. Men vi har även en möjlighet att få en direkt koppling mellan direkta observationer och en mer abstrakt representation av fysiska fenomen. Man kan överföra data från bilderna till numeriska värden i tabeller eller grafer, genom att spåra positionerna och följa dem över tid. Man kan även gå den andra vägen och överföra data från modeller till positioner i videon och jämföra med verkligheten genom grafiska ”overlays”. Vi får en övergång mellan olika representationsformer vilket underlättar inläringen.

Möjligheten att kombinera videoanalys och modellering ger möjligheten att se de begränsningar som råder med de förenklade modeller som vi har i fysikböcker och genom förfina modellerna visa att det är möjligt att förklara vardagsfenomen som oftast behandlas på ett ofullständigt sätt i fysikböcker.

Denna handbok är ett försök att visa möjligheterna med videoanalys och modellering i kombination i fysikundervisningen. Tracker är ett program som uppdateras ofta, varför beskrivningen i handboken kan skilja sig lite från den senaste versionen.

*Augusti 2012*

Jonas Persson

## **Förord till revision 2.0**

När det gäller programvara sker ofta ganska stora förändringar under en kort tidsperiod. När det gäller Tracker så har programmet genomgått vissa förändringar som gör att den förra upplagan av detta häfte har blivit lite utdaterat. För att komma till rätta med detta har jag anpassat text och bilder till den senaste versionen 4.80 (i maj 2013). Den viktigaste förändringen som skett gäller tillgängligheten av videor där den inbyggda Library Browser gör det enklare att hämta videor från andra källor, samtidigt som man har lättare att hålla reda på sina egna videor.

*Mai 2013*

Jonas Persson

### **Forord til revisjon 3.0**

Denne utgaven tar utgangspunkt i Tracker versjon 6.0.0, utgitt 31. mai 2021[6].  
Informasjon om bruk av programmet er i stor grad hentet fra Tracker Help [10].

Institutt for fysikk

Jonas Persson og Sander Ruud  
*Trondheim, Juni 2021*

# Innhold

<b>1</b>	<b>Innledning</b>	<b>7</b>
<b>2</b>	<b>Video</b>	<b>9</b>
2.1	Oppløsning . . . . .	9
2.2	Avspillings- og innspillingshastighet . . . . .	10
2.3	Eksponeringstid . . . . .	10
2.4	Whitebalance . . . . .	11
2.5	Valg av kamera . . . . .	11
2.6	Tilbehør . . . . .	11
2.7	Innspilling av video . . . . .	12
<b>3</b>	<b>Videoanalyse</b>	<b>13</b>
3.1	Tracker . . . . .	13
3.2	Rask innføring i Tracker . . . . .	14
3.3	Bruke Tracker . . . . .	16
3.3.1	Menyer . . . . .	17
3.3.2	Verktøyraden . . . . .	21
3.3.3	Video . . . . .	23
3.3.4	Videoklipp . . . . .	24
3.3.5	Clip Settings . . . . .	25
3.3.6	Video Filter . . . . .	25
3.3.7	Eksportering av video . . . . .	26
3.4	Koordinatsystem og kalibrering . . . . .	27
3.4.1	Coordinate System Menu . . . . .	27
3.4.2	Akser (Axes) . . . . .	28
3.4.3	Lengekalibrering . . . . .	28
3.5	Tracks . . . . .	29
3.5.1	Å lage et Track . . . . .	29
3.5.2	Markering av Track . . . . .	29
3.5.3	Track Control . . . . .	30
3.5.4	Autotracker . . . . .	32
3.5.5	Presentasjon av data . . . . .	35
3.5.6	Tabellvinduet . . . . .	38
3.5.7	Data Builder . . . . .	39
3.5.8	Data Tool . . . . .	40
3.6	Trackerfiler (.trk, .trz) . . . . .	41

<b>4 Videomodellering</b>	<b>43</b>
4.1 Kast med ball . . . . .	43
4.2 Luftmotstand . . . . .	46
<b>5 Gravitasjonskonstanten på månen</b>	<b>49</b>
<b>6 Solflekker – Solens rotasjon</b>	<b>53</b>
<b>7 Coefficient of restitution</b>	<b>57</b>
<b>8 Kollisjoner</b>	<b>61</b>



# Kapittel 1

## Innledning

Utviklingen av kamerateknologi de siste årene har gjort at nesten alle har tilgang til et bra kamera uansett hvor de er. Prisene har gått ned og billigere, bedre, og lettere kamera gjør det enklere å stille opp eksperimenter enn det har vært noen gang før.

Med videoanalyse får man mulighet til å se eksperimenter og tilhørende grafer samtidig, eller til og med legge målepunkter oppå videoer. Ved å kunne se virkeligheten sammen med den fysiske modellen kan man frigjøre seg fra labsalen, ved å bare ha med en smarttelefon.

I dag finnes det utallige verktøy for videoanalyse. I undervisningssammenheng finnes det blant annet Verniers Logger Pro<sup>®</sup> 3 [1], WINalyze [2], ProAnalyst [3], Capstone fra PASCO [4] og Tracker. Tracker er gratis opensource software, og forfatterne har valgt å bruke Tracker som er et enkelt program å bruke, samt et program med god dokumentasjon og et stort antall artikler som beskriver eksperimenter som analyseres med Tracker.

Dette dokumentet går gjennom hele prosedyren fra innspilling av video til analyse og modellering med Tracker, samt flere eksempler på eksperimenter.



# Kapittel 2

## Video

Så godt som alle unge mellom 16 og 34 år bruker smarttelefon til privat bruk. Mange har dessuten redigert og lagt ut egne filmer på sosiale plattformer som TikTok, Instagram, Facebook og Youtube. Dette gjør at de er i langt større grad enn før er vant til å arbeide med videofilmer. Det er i dag mulig å spille inn filmer med enkle hjelpemidler som smarttelefon, webkamera, eller eksterne kamera. Det finnes en del faktorer som påvirker kvaliteten til videoen, som man må ta hensyn til for å kunne analysere innholdet. Dette gjelder oppløsning, innspillingshastighet, avspillinghastighet, eksponeringstid, whitebalance, aperture, stabilitet, fokus, aspect ratio, og filstørrelse.

Det anbefales å ha prøvd ut innstillinger før filming begynner for fullt. Vi anbefaler en applikasjon laget av Camerasim, som gir deg muligheten til å prøve ut diverse fotoinnstillinger [5].

### 2.1 Oppløsning

Oppløsningen til et bilde fra et digitalkamera er begrenset av sensoren som gjør om lys til elektriske signaler (måler lysstyrken).

Sensoren består av et antall piksler, som i prinsippet forteller hvor mange fotoner som treffer sensoren. Oppløsningen oppgis som antall piksler i bredde og høyde. F. eks inneholder et bilde med 640 x 480 piksler 307 200 piksler.

Oppløsningen til et bilde gir en øvre begrensning på hvor mye informasjon et bilde kan ha. Det er ikke alltid slik at å øke oppløsningen automatisk vil gjøre bildet bedre, tvert i mot kritiseres mange digitalkamera og smarttelefoner for å reklamere for unødvendig høy oppløsning uten at det gir utslag for hva objektivet kan levere.

På samme måte som digitalkameraer har oppløsning, gis også det en oppløsning for video. Verdt å nevne: 720p, 1080p, 4k, 8k.

Større oppløsning resulterer i større videofiler. Selv om moderne videokamera ofte kan filme i høye oppløsninger er ikke dette alltid nødvendig. Ofte kan du velge bort en høyere oppløsning til fordel for f. eks høyere innspillingshastighet.

I forbindelse med oppløsning er det også verdt å nevne zoom. Det er hovedsakelig to typer zoom, digital og optisk zoom. Digital zoom er i prinsippet kun å skjære bildet, mens du samtidig beholder samme sideforhold (aspect ratio). Dette gjøres elektronisk, så det øker ikke oppløsningen på bildet, men med moderne

software gjør ofte et forsøk på å retusjere området eller lyden som kommer fra dette stedet. Optisk Zoom benytter seg av mekaniske innretninger for å bevege linsene på et kamera, som bøyer lyset på en bestemt måte som resulterer i zoom uten å endre oppløsningen på bildet.

## 2.2 Avspillings- og innspillingshastighet

Innspillingshastigheten forteller hvor mange bilder som tas hvert sekund av kameraet. Når vi ser på, eller redigerer opptaket kan vi velge avspillingshastigheten – antall bilder som vises på skjermen per sekund: fps - fra- per second. Holder du bildekvaliteten konstant vil naturligvis filstørrelsen bli større jo høyere fps du tar bruker.

Ved å senke avspillingshastigheten vises filmen i sakte film (Slow motion), mens hvis vi øker den spilles det av raskere. 24 bilder i sekundet er standard for filmer, mens 30 fps og 60 fps brukes ofte i digitale videoer.

Når vi ser på en skjerm er den begrenset av refresh rate – antall ganger skjermen klarer å oppdatere bildet den viser. Dette måles i Hz. Dette betyr i prinsippet at du ikke vil kunne se høyere fps enn refresh-raten, uansett hvor mye kameraet kan ta opp (merk at det fortsatt finnes egenskaper ved høyere fps, som stabilitet, som gjør det ettertraktet likevel). Merk at selv om skjermen vi ser på er begrenset av en avspillingshastighet, vil vi fortsatt i Tracker kunne ha nytte av høyere fps.

## 2.3 Eksponeringstid

Eksponeringstiden forteller hvor lenge det samles inn lys til et bilde. Det er et velkjent problem i fotografering, der man tilpasser eksponeringstiden (og filmen) til det man skal fotografere. Her snakker vi om eksponeringstider ned til 1/1000 sekund. Det er stor forskjell på utstyr som trengs for å filme formel 1 til det som er nødvendig for å filme en student som gjør en øving.

Eksponeringstiden settes ofte automatisk av digitalkamera, men man kan kontrollere dette manuelt hvis ønskelig. Innstillingene vil variere fra kamera til kamera, men i praksis innebærer dette ofte bare å velge mellom Auto, eller et manuelt program, hvor man selv velger eksponeringstid, blendertid, og ISO. For våre formål er det nyttig å låse eksponeringstiden og endre på alt det andre.

Hvis vi ser på et eksempel med et objekt som beveger seg med 10 m/s, og antar at objektet er 10 cm langt. Hvis eksponeringstiden er på 1/60 sekund, vil objektet da bevege seg ca. 16 cm, det betyr at det vil se ut som objektet er ca. 26 cm langt, som gjør analysen ubrukelig. I dette tilfellet trenger vi en eksponeringstid på under 1/500 sekund. Dette er i seg selv ikke et problem, men merk at vi da trenger godt med lys. Hvis ikke vil kameraet kompensere for det dårlige lyset ved å skru opp eksponeringstiden.

Merk at det går an å øke ISO (lysfølsomheten), som vil gjøre bildet lysere, men det vil gjøre bilde mer kornete (merk! Selve oppløsningen til bildet endres ikke, men kvaliteten på bildet gjør at det vil se slik ut). Dette er en form for elektronisk forsterkning som øker støyen i bildet.

## 2.4 Whitebalance

Whitebalance er en innstilling som tilpasser bildet etter fargetemperatur. Ulikt lys har ulik fargetemperatur, og hensikten er at hvitt på motivet skal være hvitt på bildet. Menneskets øye er i stand til å gjøre dette automatisk, men det kan ikke alle kameraer. Det gjøres ofte en god jobb automatisk justert av kameraet, men den sliter spesielt i sammenhenger hvor det kombineres ulike typer lys, f. eks dagslys og utelampe. Du legger fort merke til om et bilde er farget feil, det ser unaturlig ut, kanskje spesielt mye av én farge.

Det er ofte to måter å justere hvitbalanse: låse den manuelt etter en referanse, eller sette en verdi. Førstnevnte er det mest nøyaktige, da velges et punkt med kjent farge (ofte hvit eller grå). Verdien er mål for fargetemperatur, som måles i Kelvin (K), ofte i intervallet 1000K–10000K. For eksempel er lave verdier fargetemperaturen til stearinlys, som gir rødlig lys. Store verdier gir blåaktig lys, som du kan få når det er overskyet.

## 2.5 Valg av kamera

Man kan kanskje få inntrykket av at det er nær umulig å spille inn gode nok videoer. For de aller fleste tilfellene er ikke dette tilfellet! For videoanalyse i fysikk vil de fleste smarttelefoner være mer enn bra nok, spesielt om man selv kan stille inn eksponeringstiden.

Hvis man vil bruke vanlige digital- eller videokameraer er dette også mulig.

Det viktigste er å tilpasse kameraet til formålet. Stort sett vil problemene vi skal løse være mulige å løse med alle kameratyper.

Du kan alltid kjøpe kameraer i butikk for noe spesifikt til det du skal gjøre. Høy innspillingshastighet er ofte dyrt i kombinasjon med høy oppløsning. Hvis du godtar lavere oppløsning (480 x 480) kan dette fås tak i billig.

Merk at smarttelefoner ofte har andre applikasjoner som kan lastes ned, der du kan styre ISO, lukkertid osv... Det viktigste er at innspillingshastigheten er konstant, slik at forskjellen i tid mellom bildene faktisk er konstante, da dette er tidskalibreringen i videoen (som vi trenger i analysen).

## 2.6 Tilbehør

Selv med et godt kamera er lurt med bra tilbehør! Et stabilt stativ der kameraet kan monteres er nyttig. Dette gir stabile videoer som blir mye enklere å analysere.

Da man ofte trenger korte eksponeringstider trenger man godt med lys. Vanlig lys på lab er et billigere alternativ til egne fotolamper, men det kan gi en kunstig farge til videoen. Hvis det fortsatt er mulig å se objektet du skal analysere gjør ikke det nødvendigvis noe.

Det finnes mange eksterne applikasjoner for smarttelefoner som gir tilgang til mange innstillinger. Det er verdt å prøve ut noen hvis du savner noen viktige funksjoner i den vanlige kamera-appen, f. eks hvitbalanse, ISO, og eksponeringstid.

## 2.7 Innspilling av video

Det er ikke vanskelig å spille inn video, men det som kan være en utfordring er at kvaliteten på videoen ikke er tilstrekkelig for en analyse. Det er viktig at man kontrollerer om videoen er brukbar før man spiller inn mange. Studenter gjør ofte feilen å spille inn alt på en gang, og må dermed spille inn alt på nytt.

Det er viktig at man har så mye lys som mulig. Det er bedre med for mye enn for lite. Man må også ha med en linjal eller noe annet i bildet man kjenner lengden til. Dette brukes som en referanse så man kan kalibrere en lengdeskala.

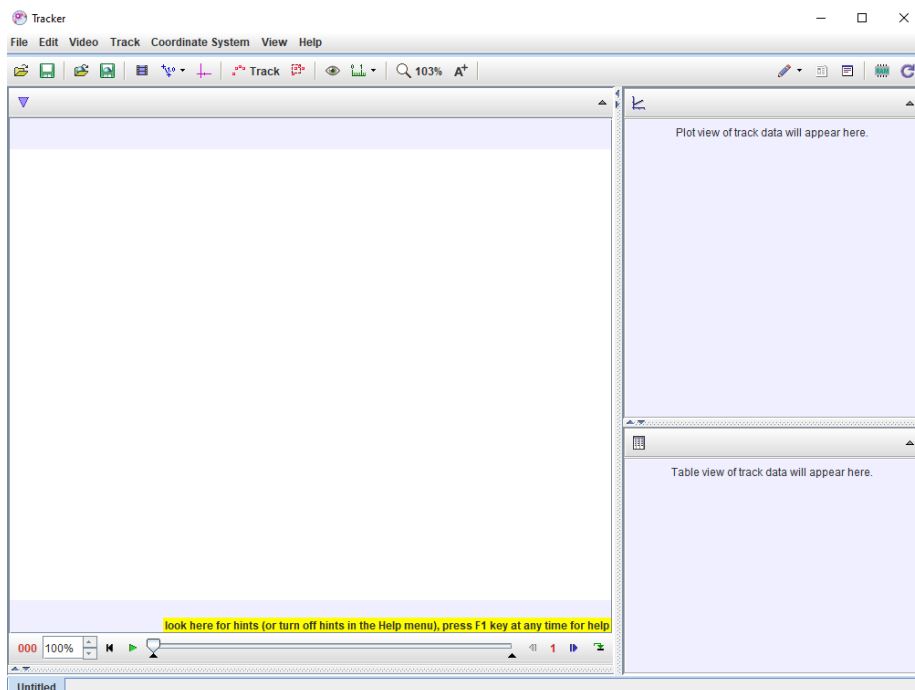
Se kapitlet om koordinatsystem og kalibrering, seksjon 3.4.

# Kapittel 3

## Videoanalyse

### 3.1 Tracker

*Tracker - Video Analysis and Modelling Tool* [6], er et gratis verktøy for bilde- og videoanalyse, og modellering, designet for bruk i fysikkundervisningen. Programmet er skrevet i *Java* med åpen kildekode [7], basert på bibliotek av Open Source Physics [8]. Tracker kan lastes ned fra <https://tracker.physlets.org/>, med mer informasjon om installasjon og kompatible versjoner av Windows, MacOS, og Linux [9]. Nytt fra 2021 er også *Tracker Online*, som er identisk med vanlig **Tracker**, men oversatt til *Javascript* og fungerer i nettleseren [13].





Figur 3.1: Startbildet i Tracker.


## 3.2 Rask innføring i Tracker

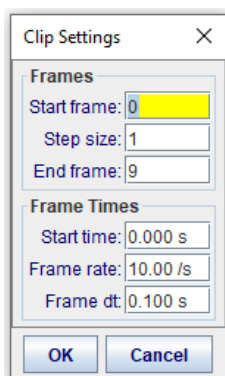
Her begynner vi med en rask introduksjon til bruk av Tracker, for mer informasjon se senere kapitler. Det antas at brukeren har installert tracker. Når du starter Tracker får du opp et vindu slik som i figur 3.1.

### 1. Åpne en video

Trykk på  i verktøyfeltet, eller via filmenyen, og velg en lokal video. Det er også mulig å hente tilgjengelige videoer fra internett gjennom  . Spill av videoen for å se at det er riktig video.


### 2. Valg av bilderuter

Alle bilderuter er ikke interessante, og man må velge de som skal brukes. Trykk på  for å se innstillingene i videoen i Clip selector (se underseksjon 3.3.5 ). I figur 3.2 ser vi valgene vi har for å velge bilderuter. Man markerer hvilken bilderute som er første og siste som skal analyseres. Hvis det er mange bilderuter kan det være nyttig å bruke en høyere steglengde enn 1. Her settes også tidsskalaen, som brukes i analysen videre.




Figur 3.2: Clip inspector

### 3. Kalibrering av lengdeskalaen

Tracker trenger en referanselengde for å vite hvor langt unna hverandre to punkter på et bilde er. Trykk på  og velg *calibration stick*. Trykk igjen for å skjule kalibreringsverktøyene. Dra de to endene til et objekt du kjenner lengden til, for eksempel en meterstav:



### 4. Valg av referansesystem og origo

Trykk på  for å vise koordinataksene, og trykk igjen for å skjule dem. Klikk og dra for å flytte origo. For å endre vinkel, dra på *x*-aksen.

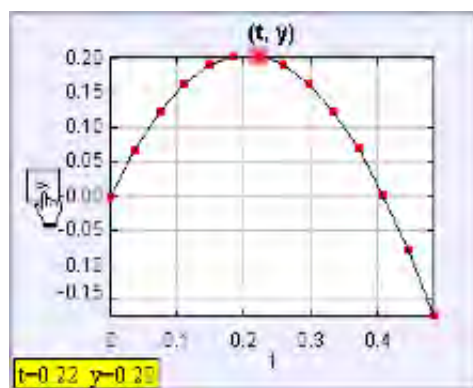


## 5. Markere objekt

Start med å lage et spor (Track), ved å trykke på **Create** for å velge ønsket type. For et objekt brukes ofte *PointMass* (Punktmasse). For å lage et spor må man markere objektet i alle bilderutene, enten manuelt, eller ved hjelp av *Autotracker* (jf. underseksjon 3.5.4). For å gjøre dette manuelt holdes shift nede mens man trykker på et objekt.—

## 6. Behandling av data

Samtidig som man markerer Tracks, vises data i tabell- og grafvindue. Trykk på et variabelnavn for å velge hvilken variabel som skal vises, se figur 3.3.




Figur 3.3: Grafvindu.

Ved å høyreklikke får man opp en meny med funksjoner, se figur 3.4



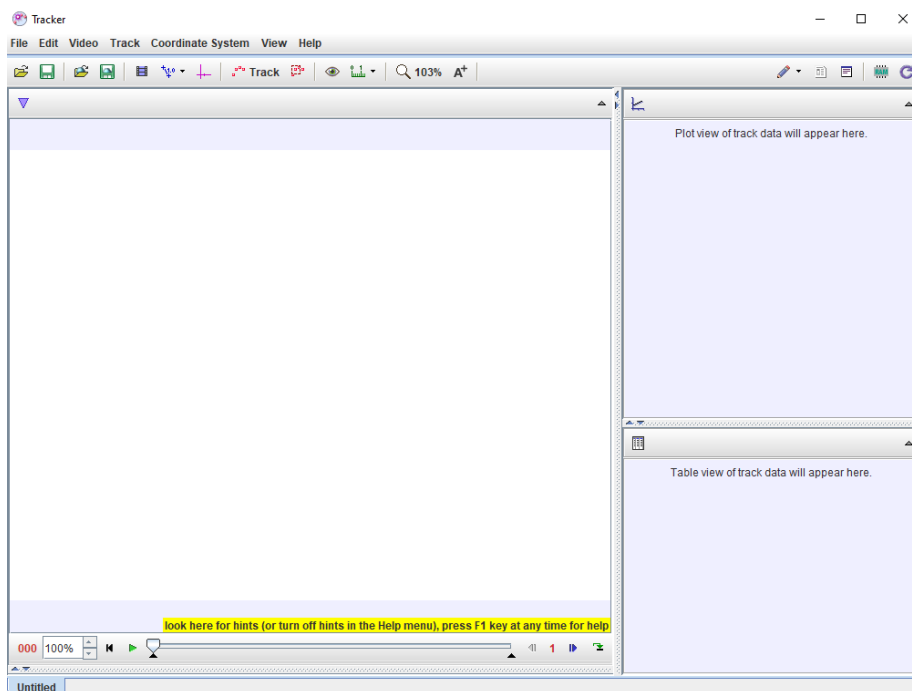
Figur 3.4: Meny for grafvinduet.

Alternativer vi kommer til å ta for oss her er *Define*, der du selv kan definere nye variabler for grafer og tabeller (jf. underseksjon 3.5.7), og *Analyze*, der vi finner blant annet statistikk og kurvetilpassning (jf. underseksjon 3.5.8).

7. **Lagre og eksportere data** Data kan lagres som en tracker-fil (*.trk*), gjennom å trykke på , eller gjennom filmenyen (jf. figur 3.3.1). Dette filformatet inneholder all informasjon for å fortsette med analysen videre i Tracker. For mer informasjon om Trackers egne filtyper se seksjon 3.6. Det er også mulig å høyreklikke og velge **Copy Data**, slik at det kan limes inn som vanlig.

### 3.3 Bruke Tracker

I forrige delkapittel ble det gitt en rask innføring i bruk av Tracker. Nå går vi grundigere gjennom hvilke muligheter som finnes i programmet. Vi har det samme startbildet som før, se figur 3.5. Vi har en rekke menyer, og en verktøylinje med de vanligste funksjonene. Vi har ett vindu for å vise video, og til høyre to vinduer som viser grafer og tabeller. Tips vises i gul tekst.



Figur 3.5: Startbildet i Tracker.

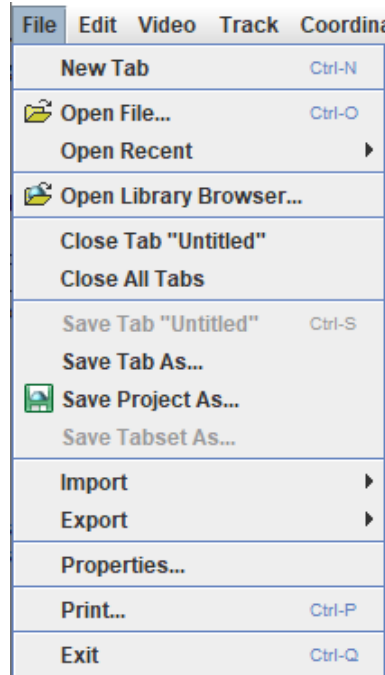
### 3.3.1 Menyer

De ulike menyene inneholder kommandoer, og her presenteres et utvalg av dem:

*File:*

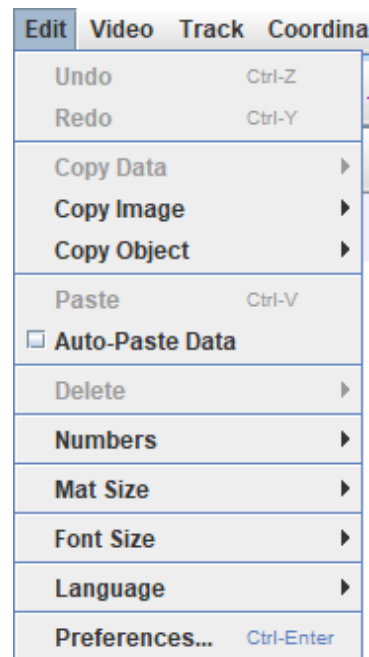
- **New Tab** åpner en ny økt.
- **Open File** åpner en video- eller trackerfil.
- **Open Recent** åpner nylig brukte filer.
- **Open Library Browser** lar brukeren gå gjennom en rekke med ressurser.
- **Close Tab** lukker økten med gitt navn.
- **Close All Tabs** lukker alle åpne økter.
- **Save Tab** lagrer
- **Save Tab As** lagrer økten med systemets egne filvindu.
- **Save Project As** lagrer filene i prosjektet som en tracker zip-fil (*.trz*, jf. seksjon 3.6).
- **Save Tabset As** lagrer de aktive fanene i prosjektet som en tracker tab-fil (*.trk*, jf. seksjon 3.6).
- **Import** importerer video eller elementer fra en tracker-fil til den aktive økten.
- **Export** eksporterer video eller tracker-fil i den åpne økten.
- **Properties** viser egenskapene for økten, som filnavn, filplassering, informasjon om videoen, og metadata.
- **Print** skriver ut vinduet.
- **Exit** avslutter programmet.

Figur 3.6: Filmenyen i Tracker.



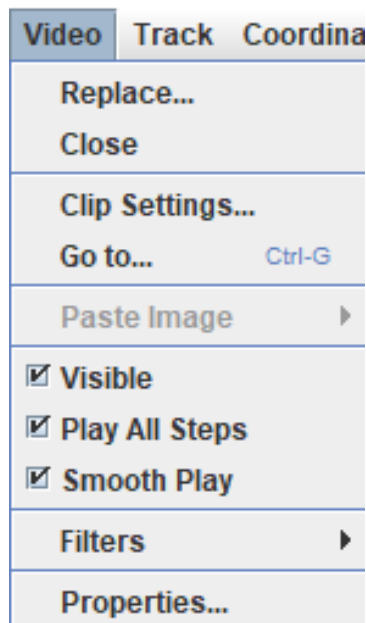
*Edit:*

- **Undo** angre siste handling
- **Redo** gjør om.
- **Copy Data** kopierer markert data til utklippstavlen.
- **Copy Image** kopierer markert vindu som et bilde til utklippstavlen.
- **Copy Object** kopierer koordinat-system eller video.
- **Paste** limer inn fra utklippstavlen.
- **Auto-Paste** limer inn fra utklippstavlen så fort det skjer en endring i utklippstavlen.
- **Delete** sletter objekter.
- **Numbers** lar deg endre enheter og størrelser.
- **Mat Size** endrer størrelsen på det hvite området rundt videoen.
- **Font Size** endrer skriftstørrelsen i hele programmet.
- **Language** angir foretrukket språk.
- **Preferences** angir innstillinger.



*Video:*

- **Replace** erstatter markert video.
- **Close** lukker markert video.
- **Clip Settings** åpner Clip Settings.
- **Go to** lar deg velge en frame eller et tidspunkt i videoen du vil gå til.
- **Paste Image** limer inn bilde fra utklippstavlen.
- **Visible** viser videoen når markert (default)
- **Play All Steps** sørger for at alle bilderuter spilles av.
- **Smooth Play** øker hastigheten or videoavspilling.
- **Filters** aktiverer ulike filter for bildebehandling, se under.
- **Properties** viser videoens egenskaper, slik som under *File*.



*Video/Filters* inneholder en mengde videofilter. For en komplett liste over tilgjengelige filter og deres funksjon, se 3.3.6. **Tracks** og **Coordinate System** behandles separat i egne kapitler, se hhv. seksjon 3.5 og seksjon 3.4.

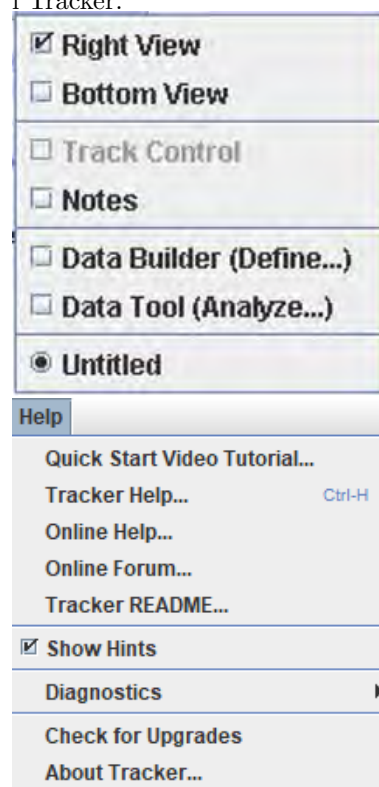
#### View

- **Right View** legger tabell/graf til høyre for videovinduet.
- **Bottom View** legger tabell/graf under videovinduet.
- **Track Control** viser kontrollvinduet for **Track**, se seksjon 3.5.
- **Data Builder (Define...)** viser **Data Builder** vinduet, underseksjon 3.5.7.
- **Data Tool (Analyze...)** viser **Data Tool**-vinduet, et enkelt analyseprogram, se underseksjon 3.5.8.

#### Help

- **Quick Start Video Tutorial** åpner en Youtube-video av forfatteren Douglas Brown, som viser et enkelt eksempel med en fallende ball.
- **Tracker Help** åpner hjelpevinduet
- **Online Help** åpner nettstedet for hjelp, med mulighet for ekstern oversettelse fra originalutgaven på engelsk.
- **Online Forum** åpner forumet til tracker på *comPADRE* sine nettsider.
- **Tracker Readme** åpner Tracker informasjonsfil.
- **Diagnostics** inneholder diverse diagnostiske hjelpemidler, som **Message Log** med logg fra java-engine om feilmeldinger, warnings, og lignende, **Start Log** gir mulighet for å opprette en logg, **Preferences file** gir xml-formatet på nåværende innstillinger, og **About Java, Xuggle og Threads** gir informasjon om ulike versjoner av programvare og hvordan programmet kjører. **Check for Upgrades** sjekker på nettet etter nye oppdateringer, og henter nettstedet for å hente en ny versjon av Tracker.
- **About Tracker** viser informasjon om blant annet versjon, hvor programmet ligger og om det er en ny oppdatering tilgjengelig, litt om forfatterene, og den lokale filplasseringen til din versjon av Tracker.

Figur 3.7: Vindu- og hjelpemenyene i Tracker.

















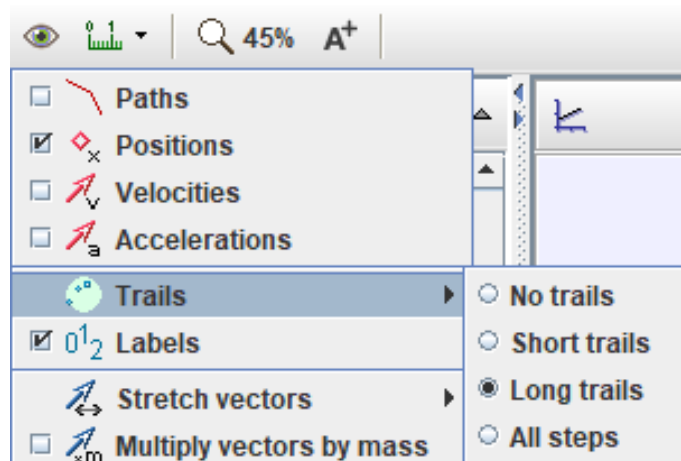
Figur 3.8: Verktøyraden i Tracker.

### 3.3.2 Verktøyraden

Verktøyraden (figur 3.8) inneholder de viktigste verktøyene man trenger for å analysere en video. Fra venstre til høyre, er verktøyene:

-  **Open** åpner en video eller tracker-fil i en ny økt.
-  **Save** lagrer økten som en tracker-fil.
-  **Open library browser** åpner OSPS digitale library browser.
-  **Save Project As** lagrer økten som en ny fil.
-  **Clip Settings** lar deg endre start-frame, end-frame, og step-size, se underseksjon 3.3.5.
-  **Coordinate Axes** lar deg vise eller skjule koordinataksene.
-  **Calibration tools** gir deg tilgang til *Calibration Stick*, *Calibration points*, og *Offset Origin*, som henholdsvis gir en stav med en kjent lengde, sette opp punkter for å kalibrere én eller to av aksene, eller definere et koordinatsystem med origo og aksene som en vanlig graf, se underseksjon 3.4.3.
-  **Track** viser *Track control*, se seksjon 3.5
-  **Autotracker** viser Autotracker, se underseksjon 3.5.4
-  **Control Track Display** gir mulighet for å velge hva som skal vises i videovinduet. Her kan du vise *positions*, som er målepunktene, eller *paths*, som lager linjer mellom ulike *positions*. Du har også mulighet for å sette opp *trails* (figur 3.9), som for hvert tidssteg viser tidligere *positions*. Dette gjør det enkelt å følge bevegelsene til et legeme du skal analysere. Vektorer som hastighet *Velocities* og *Accelerations* kan også vises, og stretches for bedre synlighet. *Multiply vectors by mass* omgjør *Velocities* og *Accelerations* til henholdsvis *Momenta* og *Net Forces*.
-  **Measuring tools** lar deg vise skjule, eller lage nye *Measuring tools*. Det er 3 typer: *Tape Measure*, som lar deg velge en lengde du vil se på, f. eks mellom to punkter, *Protractor*, som måler vinkelen på en gradskive, f. eks mellom to punkter, og *Circle Fitter*, som lar deg markere punkter og tilpasse en sirkel til punktene. Alle 3 kan brukes til målinger, og får opp egne grafer når du definerer en ny.
-  **Zoom level** lar deg forstørre videoen, der *To Fit* indikerer at hele videoen har plass i videovinduet,

- A<sup>+</sup>** **Set font and icon sizes** lar deg forstørre ikonstørrelse og skrift. Denne endringen påvirker hele programmet, ikke bare videovinduet.



Figur 3.9: Trails i Tracker.





### 3.3.3 Video

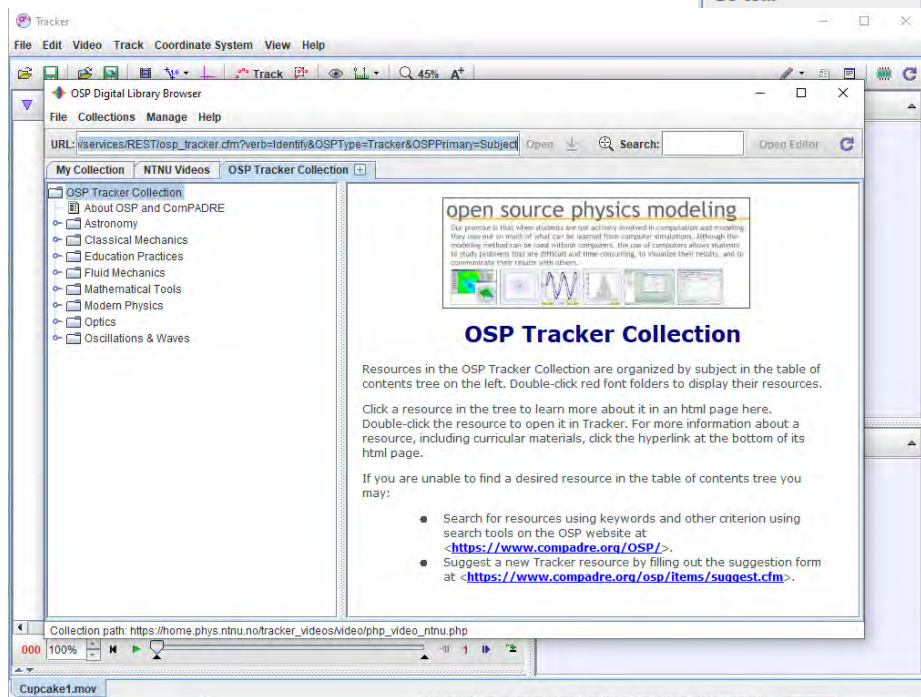
Tracker kan analysere 3 ulike videotyper [10]:

1. **digitale videofiler** som (.mov, .avi, .mp4, .flv, .wmv, .ogg). Tracker bruker *Xuggle* i desktopversjonen, og HTML5 i webversjonen. Merk at tidligere versjoner av tracker benyttet seg av *QuickTime*.
2. **animerte GIFs** (.gif).
3. **bildesekvenser**, med én eller flere digitale bilder (.jpg, .png, utklippstavlen).

#### Åpne video

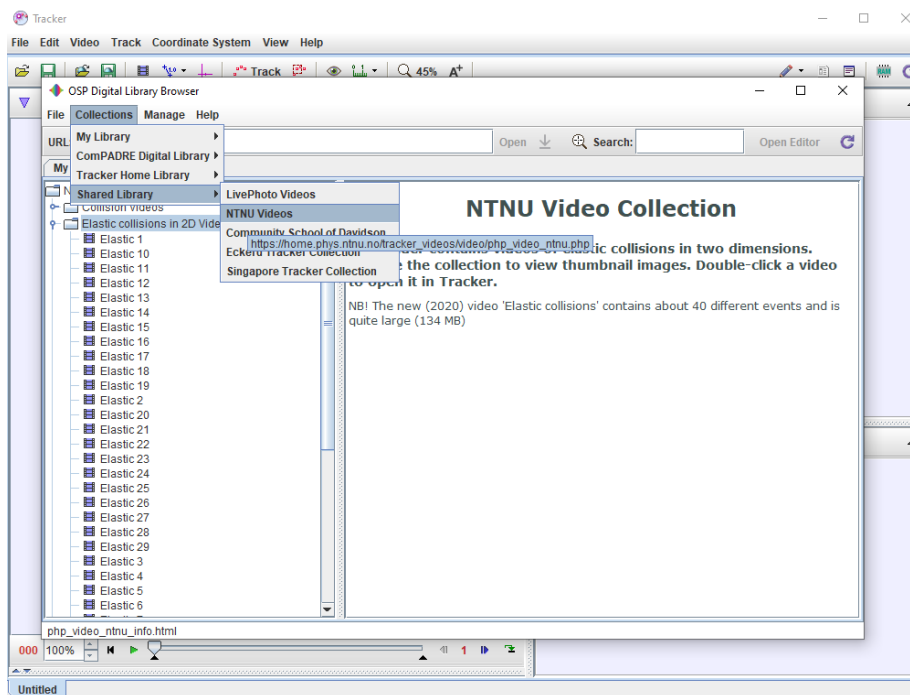
Det finnes flere ulike måter å åpne en video. Det enkleste er å dra videofilen inn i Tracker. Du kan også trykke på , eller gå via **File**, eller **Video**.

Det er også mulig å åpne en video fra nettleseren, ved å trykke på , eller gjennom **File**. Her har du tilgang til en rekke kolleksjoner av ressurser. For eksempel *Open Source Physics* sin egen Tracker collection, se 3.3.3.



Figur 3.10: OSP Tracker Collection.

Det er også mulig å lage egne brukerdefinerte collections. Bildet under viser samlingen fra NTNU, utviklet av Jonas Persson. Det er også mulig å legge til tracker-filer (.trk), og en .pdf, med ferdig analyse, som kan være gunstig i undervisningssammenheng.



Figur 3.11: NTNU Video Collection.


### 3.3.4 Videoklipp

I alle videofiler finnes det innebygd informasjon om blant annet bildeformat og innspillingshastighet. Merk at noen programmer setter innspillingshastigheten til 30 bilder i sekundet by default uten å koble det til avspillingshastigheten, så vær oppmerksom på å sjekke dette, da Tracker kun sjekker innspillingshastigheten i videofilen.

Et videoklipp er en delmengde av bilderuter i en video, kalt steg (*steps*). Disse stegene er definert av tre verdier: **Start Frame**, og **End Frame**, som definerer første og siste frame i videoklippet, og **Step Size** som angir antall bilderuter per steg. Dette er lurt å teste litt ut, det er ikke alltid første bilderute fra videoen er det beste stedet å starte. Ofte hvis det er veldig mange frames trenger du ikke alle bilderutene, og step size blir dermed et nyttig verktøy for lange videoer.

Egenskapene for et videoklipp kan bestemmes i *Video Player* eller i **Clip Settings**, (jf. underseksjon 3.3.5). Husk å sette disse egenskapene i forkant av analysen, da endring av disse egenskapene endrer dataen.

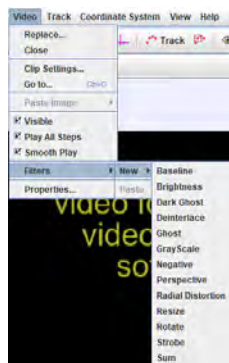
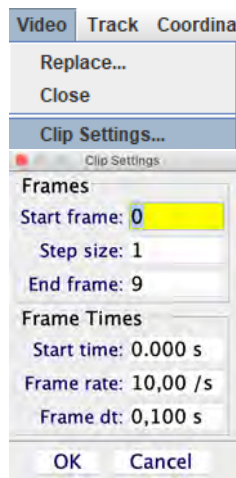
### 3.3.5 Clip Settings

**Clip Settings** viser innstillingene som det aktuelle videoklippet har. Den lar deg velge **Start Frame**, **End Frame**, og **Step Size**, og bildehastighet (*frame rate*). Standard for filmer er 24, 30, eller 60 bilder i sekundet, og hvis du har spilt inn video med høy hastighet må du endre framerate her. For å vise **Clip Settings**, trykk på , eller naviger via **Video**.

### 3.3.6 Video Filter

Ofte er det praktisk å gjøre noen endringer på videoen for analysen. Kanskje det er for å få frem detaljer bedre, eller fjerne noe uønsket, spesielt hvis videoen er av lav kvalitet.

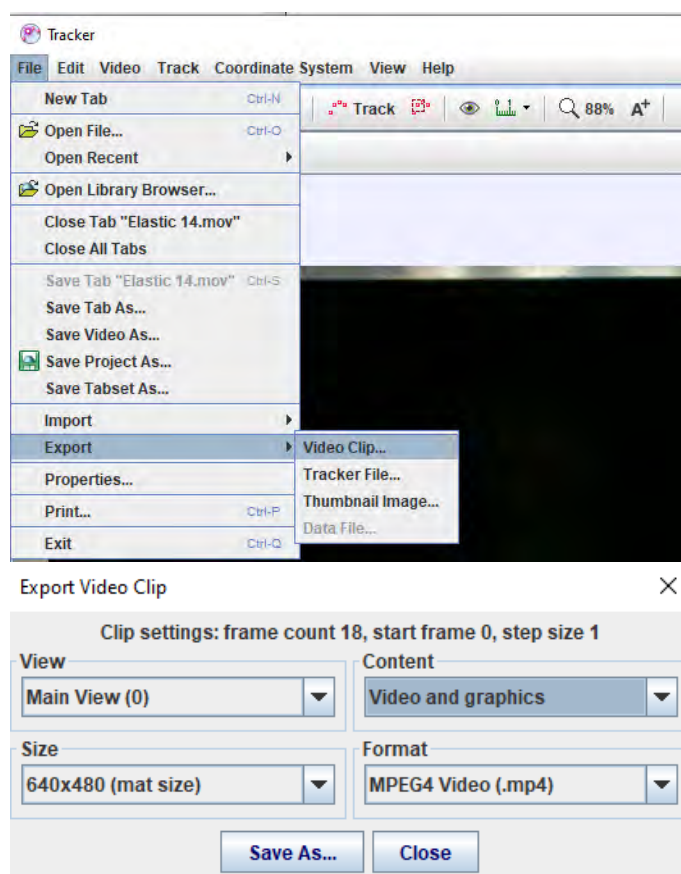
- **Baseline** fjerner et bakgrunnsbilde fra alle bilderuter.
- **Brightness** endrer lysstyrke og kontrast.
- **Dark Ghost** legger igjen spor av av objekter i bevegelse.
- **Deinterlace** fjerner doble bilder fra interlaced videos. Dette ble ofte brukt på eldre TV-er, for å spare båndbredde. Her sendte man halve bildet (rad 1, 3, 5, ...) først og deretter andre halvdel (rad 2, 4, 6, ...). Dette ble satt sammen av skjermen og førte til en effektiv dobling av avspilningshastigheten.
- **Ghost** etterlater et spor av spøkelsesbilder som blir mer og mer uklare jo lenger bak der er.
- **Grayscale** viser svart-vidt bilde.
- **Negative** viser bilde i negativ.
- **Perspective** brukes for å korrigere bilder som vises i vinkel i stedet for rett i mot.
- **Radial Distortion** brukes for å korrigere når man bruker fish-eye, eller vidvinkel på kameraet.



- **Resize** endrer dimensjonene til bildet.
- **Rotate** roterer videoen i 90-graders inkremerter.
- **Strobe** gir samme effekt som ghost, men alle sporene har lik lysstyrke.
- **Sum** summerer bilderuter, eller tar et gjennomsnitt.

### 3.3.7 Eksportering av video

Endringene man gjør i en video kan lagres ved å eksportere videoen. Her får man mulighet til å velge en rekke innstillinger for videoen som eksporteres. Her kan du velge hva som skal være med i videoen gjennom **Content**, velge oppløsning gjennom **Size**, og filformat gjennom **Format**. Har du flere vinduer å velge mellom kan du også velge vinduet som eksporteres i **View**.



Figur 3.12: Eksportering av video med tilhørende innstillinger.

## 3.4 Koordinatsystem og kalibrering

Gjennom bildehastigheten og starttid har vi en tidskalibrering – det vil si, et bilde i videoen beskriver et tidspunkt. Det som gjenstår er en sammenheng mellom et bildebunkt på skjermen og en posisjon i virkeligheten, dette gjøres gjennom **Coordinate System**.

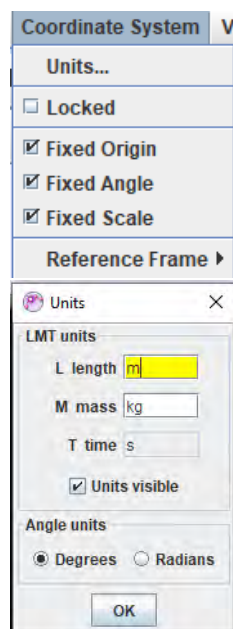
Posisjonene i bildet angis som pikselposisjon fra origo, som er satt til øverst til venstre i bildet. Positiv  $x$ -retning går mot høyre fra origo, mens positiv  $y$ -retning går nedover fra origo. For eksempel vil et 1920x1080-bilde ha øverst til venstre i punkt  $(0, 0)$  og nederst til høyre i  $(1920, 1080)$ .

Den ekte posisjonen, er den skalerte posisjonen for et punkt relativt til et spesifisert referansesystem. Vi trenger å definere origo, vinkel mellom bildets  $x$ -akse og den ekte  $x$ -aksen, og en lengdeskala, som definerer antall piksler per lengdeenhet i labratoriesystemet. Har vi definert disse, vil Tracker automatisk finne en kobling mellom virkeligheten og videoen slik at vi kan bruke ekte enheter for videoen.


### 3.4.1 Coordinate System Menu

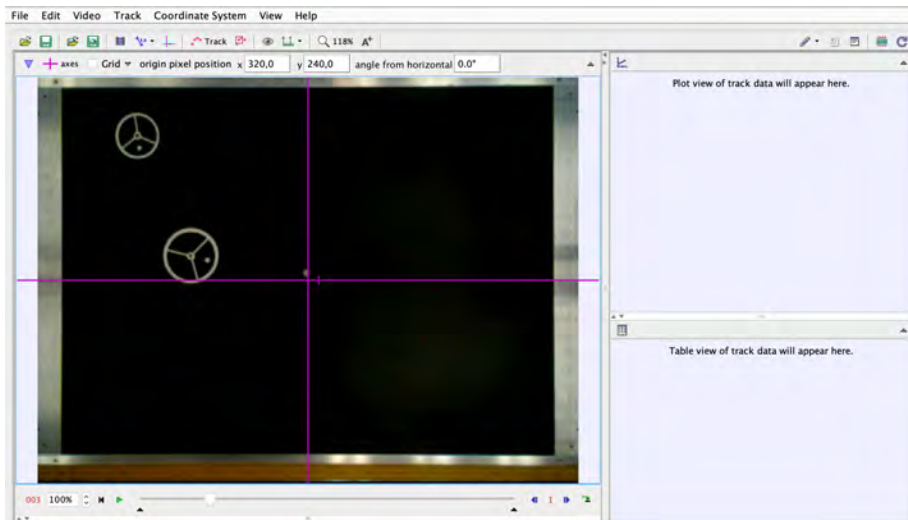
I menyen for koordinatsystemet har vi tilgang til en rekke nyttige funksjoner. **Units** lar deg definere enhetene for lengde, masse, tid, og vinkel for koordinatsystemet. **Locked** hindrer at det skjer noen endringer for skalering, origo og vinkel. **Fixed Origin, Angle og Scale** angir at det ikke er noe som endrer seg fra bilde til bilde.

**Reference Frame** gjør at man kan la koordinatsystemet følge et markert objekt eller massesentrum i videoklippet. Dette er spesielt anvendelig når man skal studere kollisjoner, se eksempel på dette i chapter 8. Merk at vinkel og skalering beholdes.



### 3.4.2 Akser (Axes)

Trykk på  for å åpne aksene, se underseksjon 3.4.2.



Figur 3.13: Tracker med koordinatsystem i videovinduet.

Origo kan trykkes på og dras til ønsket plassering, eller skrives inn under **origin pixel position**. Den positive  $x$ -aksen er markert med én strek. Vinkelen endres ved å trykke på  $x$ -aksen, og flytte den til ønsket vinkel. Ved å holde inne shift, kan dette gjøres i 5-graders inkremitter. Du kan også skrive inn vinkelen manuelt under **angle from horizontal**.

Man burde velge origo slik at tabeller er så lette å lese som mulig. Ofte velges nullpunktet ved start- eller slutt punktet til objektet som analyseres.

### 3.4.3 Lengdekalibrering

For å kunne bruke videoen må vi vite hvilken lengdeskala vi har i videoen. Koblingen mellom den virkelige verden og bildet gjøres med kalibreringsverktøy.

På verktøysraden, har du tilgang til 3 kalibreringsverktøy .

**Calibration Stick** har en bestemt lengde og vinkel, hvor standard er en horisontal meterstav. Ved å trykke og dra på et ytterpunkt kan du forlenge staven. Trykker du et sted på midten kan du flytte den men beholde lengden og vinkelen.

**Calibration points** lar deg velge to punkter med kjente posisjoner, så regner Tracker ut avstanden mellom dem.

**Offset Origin** lar deg velge et punkt med kjente koordinater, slik at når du flytter punktet flytter også origo relativt til dette punktet.

## 3.5 Tracks

**Tracks** i Tracker er et objekt koblet til videoen som utvikler seg over tid gjennom data man får fra å markere et objekt i videoen eller skape en matematisk modell, se underseksjon 3.5.7.

- **Point Mass** er objekter som beveger seg som punktmasser. Disse markeres manuelt, eller med **Autotracker**, se underseksjon 3.5.4.
- **Center Of Mass** gir massesentrum for en samling av punktmasser (**Point Mass**).
- **Vector** representerer en vektor, ofte brukt til å beskrive krefter.
- **Vector Sum** representerer en sum av vektorer.
- **Line Profile** er et redskap for å måle lysstyrke og RGB data langs en linje i videoen, som kan brukes til å analysere spektrallinjer.
- **RGB Region** er et redskap for å måle lysstyrke og RGB-data over tid i et sirkulært område i videoen.
- **Kinematic Model** beskriver bevegelse gjennom en analytisk modell.
- **Dynamic Model** bruker krefter til å beregne akselerasjonen til et objekt. Dette er mulig i kartesiske koordinater, polarkoordinater, eller som et tolegemeprobblem.
- **External Model** lar deg importere en tekstfil med ulike verdier. Dette kan f. eks være en liste med posisjoner i *.csv*-format.

### 3.5.1 Å lage et Track

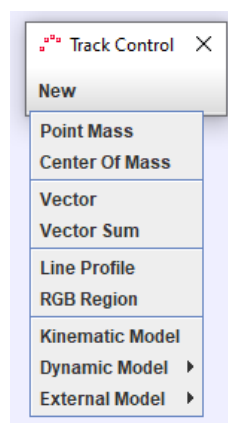
Dette gjør man når koordinatakser og lengdekalibrering er definert. For å lage **Track**, åpne **Track Control** ved å trykke på **Track** i verktøylinja. Fra **Track Control** kan du velge et nytt objekt, se figur 3.14.

Merk at når man lager en punktmasse kan man også skrive inn massen i raden under verktøysraden.

### 3.5.2 Markering av Track

Markering handler om å bestemme posisjonen for objektet i alle bilderutene i videoklippet. For punktmasser har vi mulighet til å gjøre dette manuelt, eller med en autotracker-funksjon, se underseksjon 3.5.4.

Den manuelle markeringen skjer med musepekeren. Hold inne *shift*-knappen, og flytt posisjonen med musen. Når man har valgt posisjon går Tracker videre til neste bilderute. Det er viktig å markere i alle bilderutene, da hastighet og akselerasjon ellers ikke kan beregnes. For å gjøre det enklere å markere kan du også zoome før du markerer.



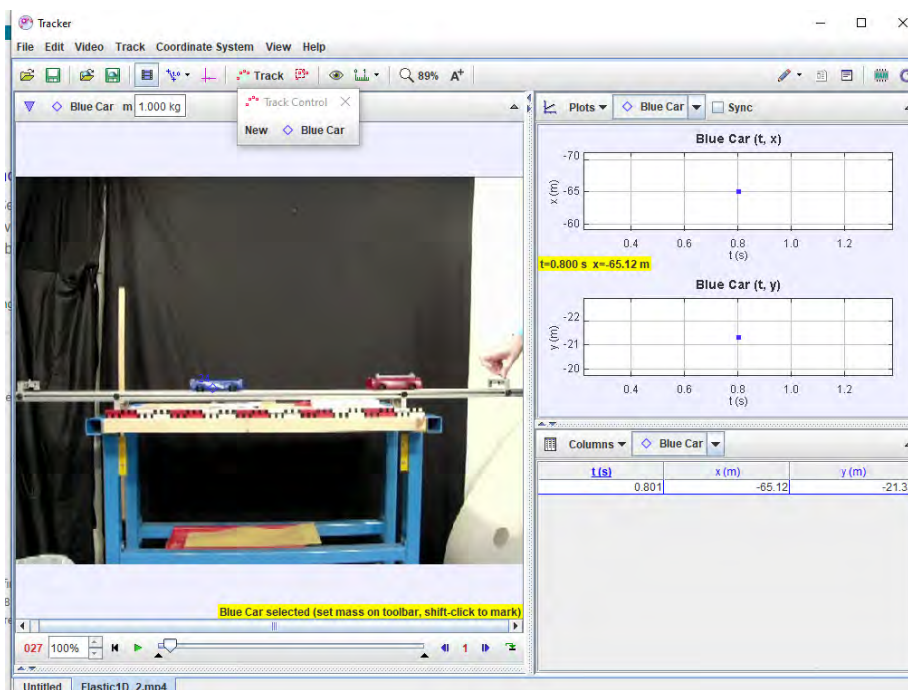
Figur 3.14: Track Control.



### 3.5.3 Track Control

Til hvert **Track** finnes det en knapp som viser navnet, fargen og symbolet til objektet. Ved å trykke på denne får man opp en meny som kalles **Track Control**. Her kan man endre egenskapene til objektet som navn, farge og symbol. **Visible** indikerer om objektet skal vises. **Locked** avgjør om det er mulig å endre objektet. **Autotracker** (jf. underseksjon 3.5.4), åpner **Autotracker** for det valgte objektet. **Define** lar deg velge parametere (f. eks masse, eller egendefinerte parametere), og datafunksjoner, som er matematiske uttrykk. Dette kan brukes til å beregne nyttige fysiske størrelser som bevegelsesmengde, trykk, osv.

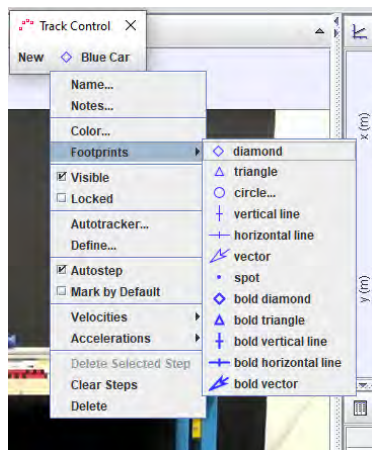
Det er lurt å velge symbol og farge slik at man kan se markeringen tydelig. I figur 3.15 har symbolet samme farge som objektet, som gjør det vanskelig å se det.



Figur 3.15: Bildet viser et objekt som er markert i en bilderute (blå firkant).

Symboler kan velges i **Track Control** under **Footprints**, se figur 3.16.



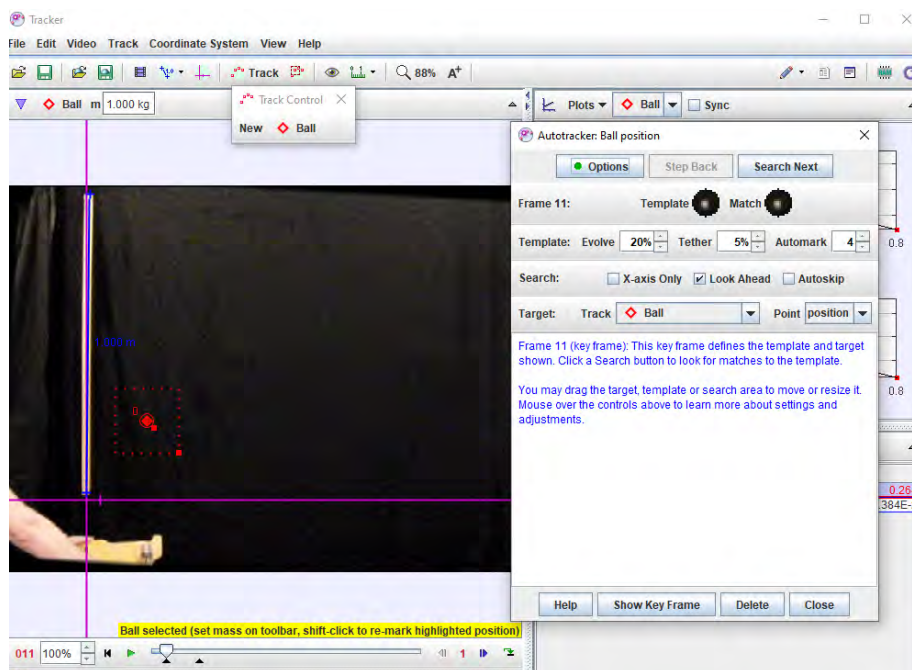


Figur 3.16: Ulike valg for Footprints.

### 3.5.4 Autotracker

I mange tilfeller kan det være et stort antall bilderuter som skal analyseres, som gjør det svært tidkrevende å markere alle rutene manuelt. Har man en video av høy kvalitet, og objektet har en konsistent form, farge, størrelse og orientering i alle bilderutene kan man anvende seg av den automatiske markeringsfunksjonen **Autotracker**. I tillegg til å slippe den manuelle markeringen blir det også mer konsistente resultater av dette hvis **Autotracker** klarer å finne objektet. Det er også mulig å markere manuelt for de rutene objektet ikke blir funnet. Det beste resultatet får man med sirkulære markører, og en hvit eller svart ring rundt et farget sentrum. Her kan man anvende ulike farger for ulike objekter, spesielt hvis objektene er veldig nærme hverandre i videoen.

**Autotracker** fungerer gjennom å definere en mal (**Template**), se figur 3.17. Hold inne *Shift+Ctrl* og marker objektet.

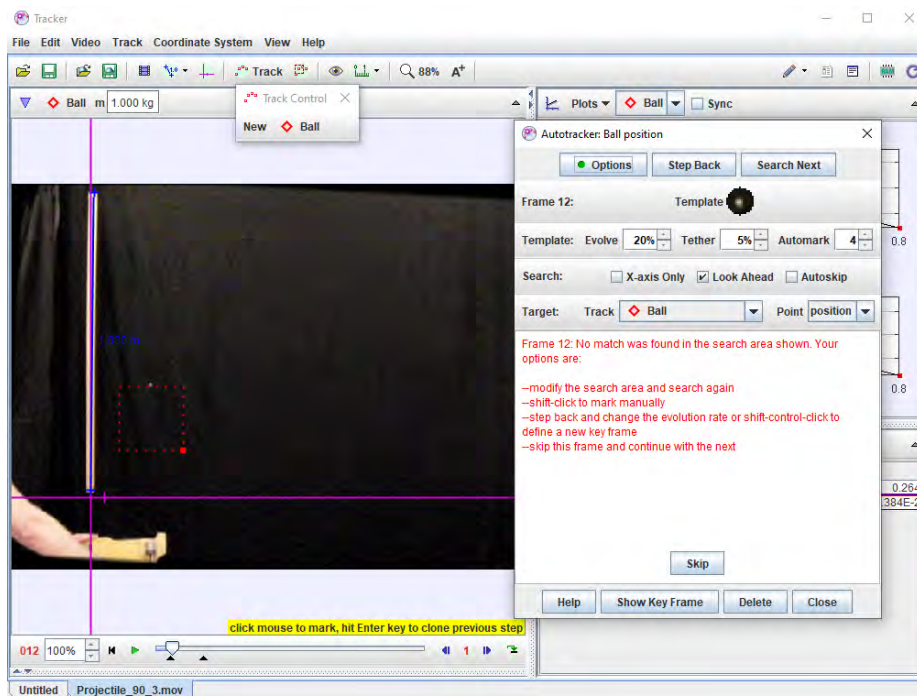


Figur 3.17: Figuren viser **Autotracker** med markert mal.

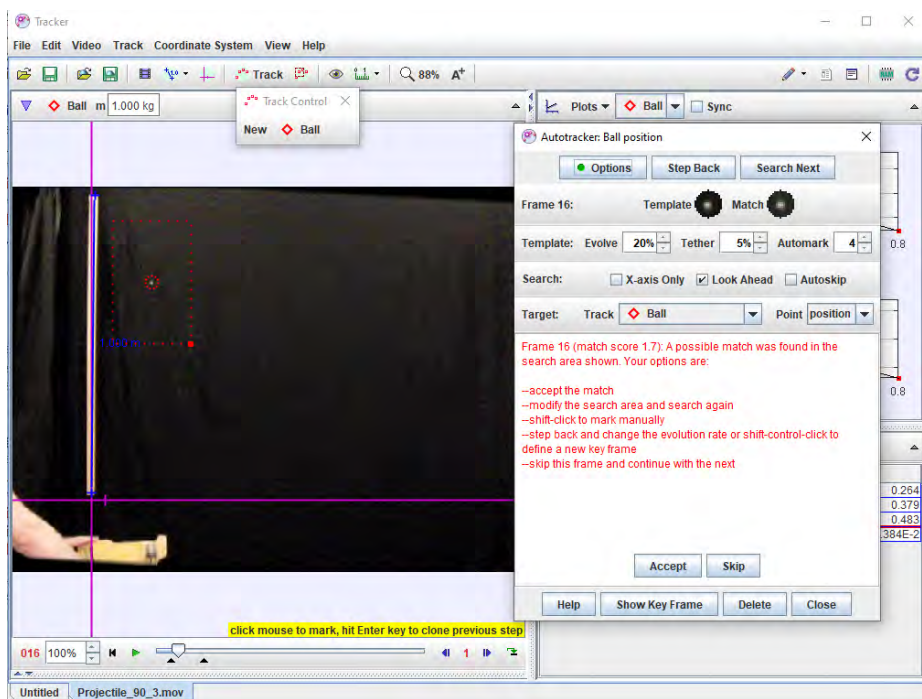
Når du har en mal kan du velge **Search Next**. Dette vil søke i neste rute etter et objekt som ligner på malen. Legg merke til firkanten rundt objektet, dette er søkeområdet **Search Area**. Hvis dette området ikke er stort nok vil ikke **Autotracker** finne objektet, se 3.18. Dette kan være et tegn på at høyere bildehastighet er nødvendig. Et mindre søkeområde gjør at **Tracker** raskere finner objektet. Det er også mulig å begrense søkingen til kun  $x$ -aksen.

Deretter kan du trykke videre så lenge **Autotracker** er sikker nok på å ha funnet objektet. Hvis treffet ikke er helt sikkert, se 3.19 kan du akseptere, markere objektet manuelt, eller hoppe over bildet. Du kan også gå tilbake til forrige bildet gjennom **Step Back** og endre **Evolution Rate**.

Autotracker finner ikke alltid objektet uten å endre innstillinger. To eksempler på dette er figur 3.18 hvor søkeområdet er for lite, og figur 3.19 hvor det potensielle funnet ikke er identisk med template.



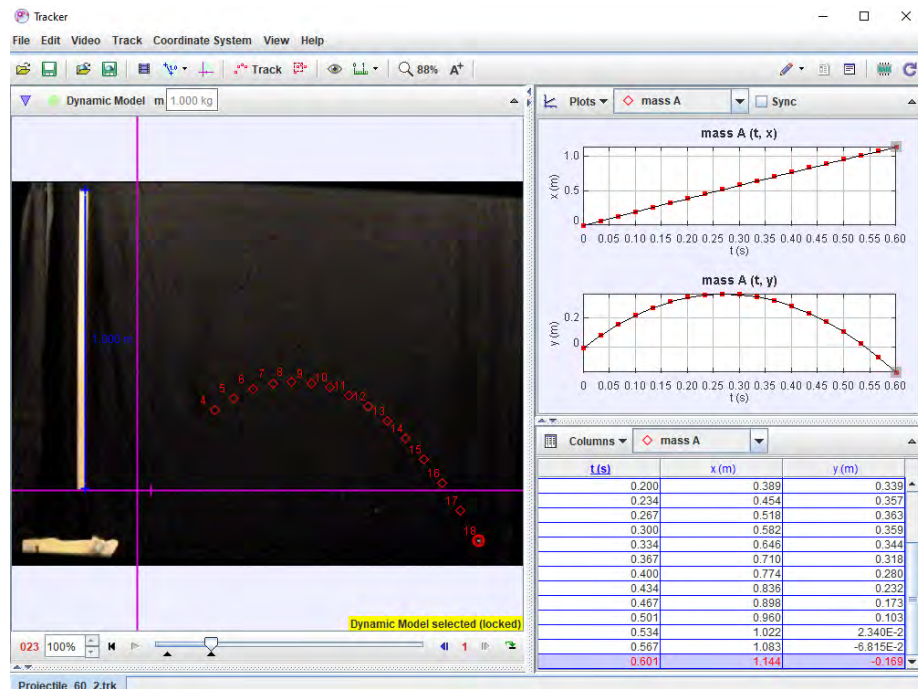
Figur 3.18: Figuren viser **Autotracker** med for lite søkeområde.



Figur 3.19: Figuren viser **Autotracker** hvor Tracker er usikker på om objektet fundet.

### 3.5.5 Presentasjon av data

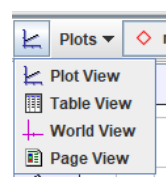
Gjennom markeringen av et **Track** lagres verdier for  $x$ - og  $y$ -koordinater. Disse datapunktene kan presenteres på ulike måter som en graf eller i en tabell. Presentasjonen gir ulike vinduer. Her kan man velge om man vil ha disse til høyre eller under videovinduet i Window-menyen, for et eksempel på dette se figur 3.20.



Figur 3.20: Bildet viser et eksempel på presentasjon av data som ved bruk av både en graf og en tabell.

Du kan også velge hva som skal vises, se 3.21. **Plot View** viser grafer, **Table View** viser tabeller, **World View** viser videoen, og **Page View** lar deg åpne nettsider fra *.html*-filer.

Det er også mulig å endret størrelsen på et vindu ved å dra det til siden. Vi skal nå gå gjennom graf- og tabellvinduet mer nøye.

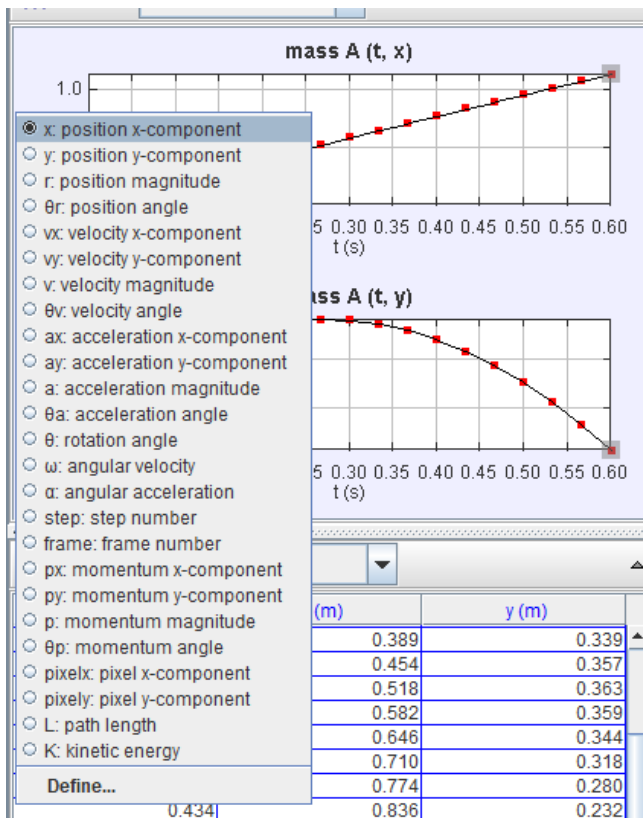


Figur 3.21: Valg av visning av data.

### Grafvinduet

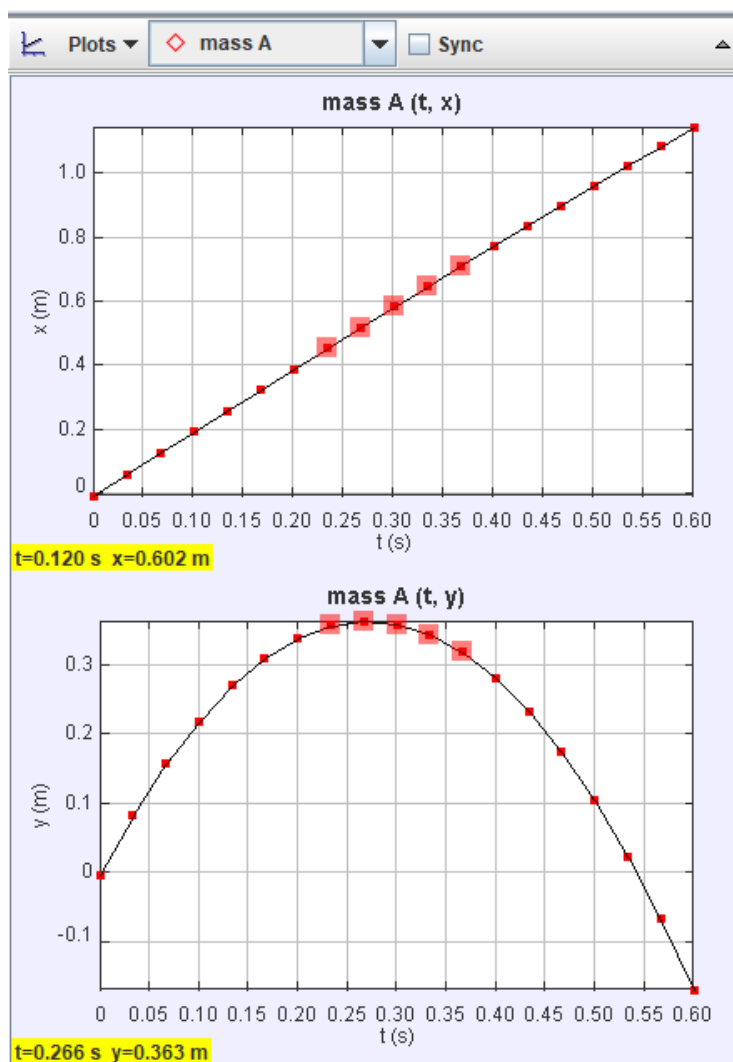
Vinduet viser en graf av valgt data. Her er det også mulig å holde over grafen, så vil koordinatene du holder over vises nederst til høyre.

Ved å trykke på akseenhetene kan man også endre hvilke verdier som skal vises i plottene eller tabellene. **Tracker** lar deg velge mellom et utvalg allerede definerte verdier se figur 3.22, eller du kan definere relasjoner selv gjennom en formel på samme måte som i Data Builder, se underseksjon 3.5.7.



Figur 3.22: Grafvinduet med ulike valg for enheter som skal vises på aksene.

Ved å høyreklikke på grafen får du mange muligheter for hvordan grafen skal vises. **Tracker** vil automatisk forstørre slik at alle punkter vises, men dette kan også gjøres manuelt. Det er også mulig å markere punkter. Disse punktene blir da også markert i andre grafer, se 3.23. Legg merke til koordinatene som vises.



Figur 3.23: Grafvinduet med to grafer og punkter markert i begge grafene.

### 3.5.6 Tabellvinduet

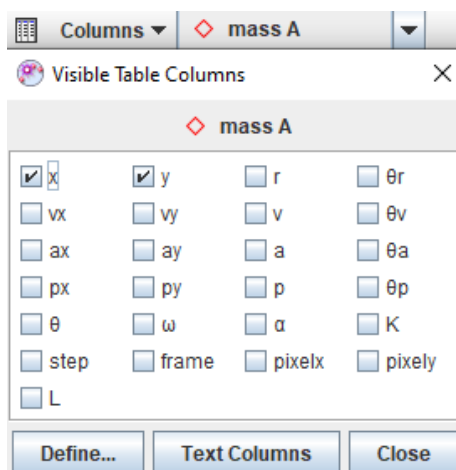
I tabellvinduet (figur 3.24) vises en tabell med data for et **Track**. Her er det også mulig å velge hvilket **Track** som skal vises. Det er mulig å kopiere data fra tabellene til et annet program eller å analysere det med **Data Tool**, se underseksjon 3.5.8.



t (s)	x (m)	y (m)
0.000	-6.116E-3	-3.058E-3
0.033	6.032E-2	8.202E-2
0.067	0.127	0.156
0.100	0.192	0.218
0.133	0.258	0.269
0.167	0.323	0.310
0.200	0.389	0.339
0.234	0.454	0.357
0.267	0.518	0.363
0.300	0.582	0.359
0.334	0.646	0.344
0.367	0.710	0.318
0.400	0.774	0.280
0.434	0.836	0.232
0.467	0.898	0.173
0.501	0.960	0.103
0.534	1.022	2.340E-2
0.567	1.083	-6.815E-2
0.601	1.144	-0.169

Figur 3.24: Tabellvinduet i Tracker.

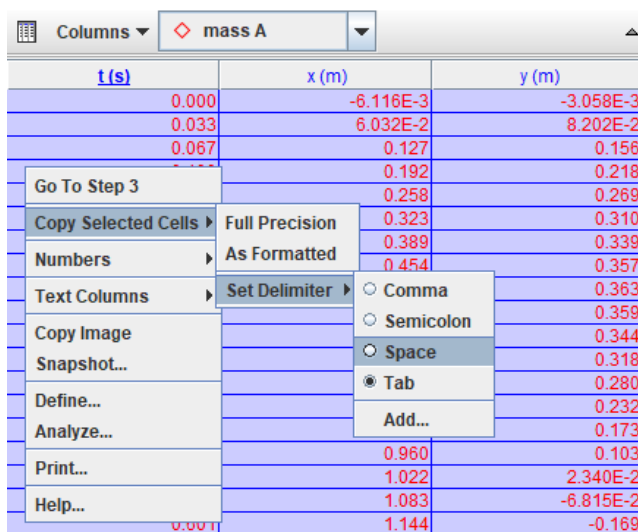
Det som presenteres er tiden  $t$ , og  $x$ - og  $y$ -posisjoner. Ved å trykke på **Columns** kan man velge hvilke variabler som skal vises (figur 3.25), eller definere egne gjennom **Data Builder**, se underseksjon 3.5.7.



Figur 3.25: Valg av variabler i tabellvinduet.



Ved å markere data og høyreklikke kan man velge å kopiere data med valgfrie skilletegn, som kan være nyttig for å eksportere data til andre programmer, se figur 3.26.



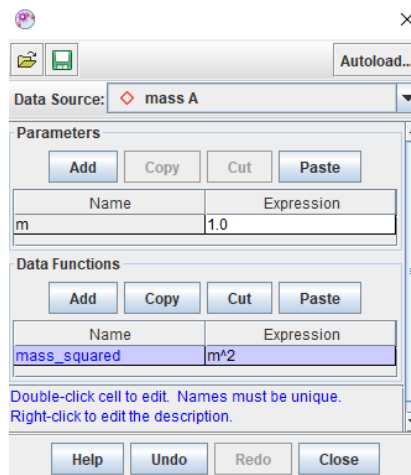
Figur 3.26: Valg av skilletegn ved kopiering av tabell.

### 3.5.7 Data Builder

**Data Builder** gjør det mulig å selv definere data og plote grafer. Det enkleste er å lage et nytt datasett i tabellvinduet. Her høyreklikker man og velger **Define**. I dette vinduet kan man legge til relevante parametere med tallverdier (f. eks masse, tyngdeakselerasjon), og funksjoner av parametere. Begge to defineres ved å trykke på **Add**. Uttrykk kan inneholde matematiske symboler som følger *OSP parser*:

- **Tall** enten i desimalform (punktum som desimaltegn), eller vitenskapelig notasjon (f. eks 3.1E-4)
- **Navn** på parametere, funksjoner, eller andre inputvariabler.
- **Konstanter** som e og pi ( $\pi$ )
- **Aritmetiske operatorer** + - \* / ^
- **Boolske operatorer** = < > ≤ ≥ <> & | !
- **Paranteser** ()
- **Matematiske funksjoner** som  $abs(x)$ ,  $acos(x)$ ,  $acosh(x)$ ,  $a \sin(x)$ ,  $a \sinh(x)$ ,  $atan(x)$ ,  $atanh(x)$ ,  $atan2(x, y)$ ,  $ceil(x)$ ,  $cos(x)$ ,  $cosh(x)$ ,  $exp(x)$ ,  $frac(x)$ ,  $floor(x)$ ,  $int(x)$ ,  $log(x)$ ,  $max(x, y)$ ,  $min(x, y)$ ,  $mod(x, y)$ ,  $random(x)$ ,  $round(x)$ ,  $sign(x)$ ,  $\sin(x)$ ,  $\sinh(x)$ ,  $sgr(x)$ ,  $sar(x)$ ,  $step(x)$ ,  $\tan(x)$ , og  $\tanh(x)$
- **If-setninger** på formen  $if(a,b,c)$  hvor  $b$  returneres hvis  $a$  er sann, og  $c$  returneres ellers.

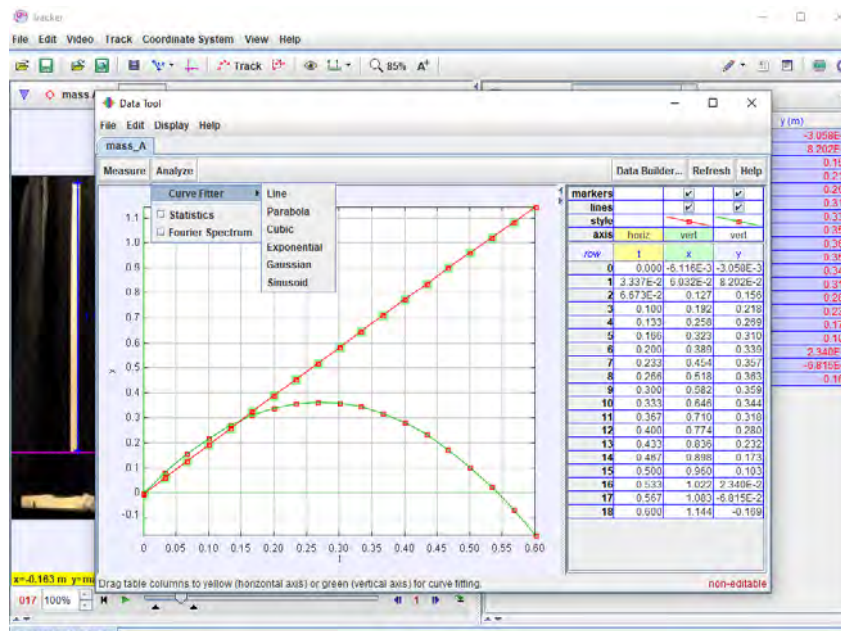
I figur 3.27 ser vi **Data Builder** for en punktmasse. Her er funksjonen `mass_squared` definert.



Figur 3.27: Vinduet til **Data Builder**.

### 3.5.8 Data Tool

Det er også mulig å analysere data med **Data Tool**. Marker data og høyreklikk og velg **Analyze**, se figur 3.28. Dette verktøyet gir tilgang til mange måter å analysere modellen, og her går vi gjennom de viktigste.



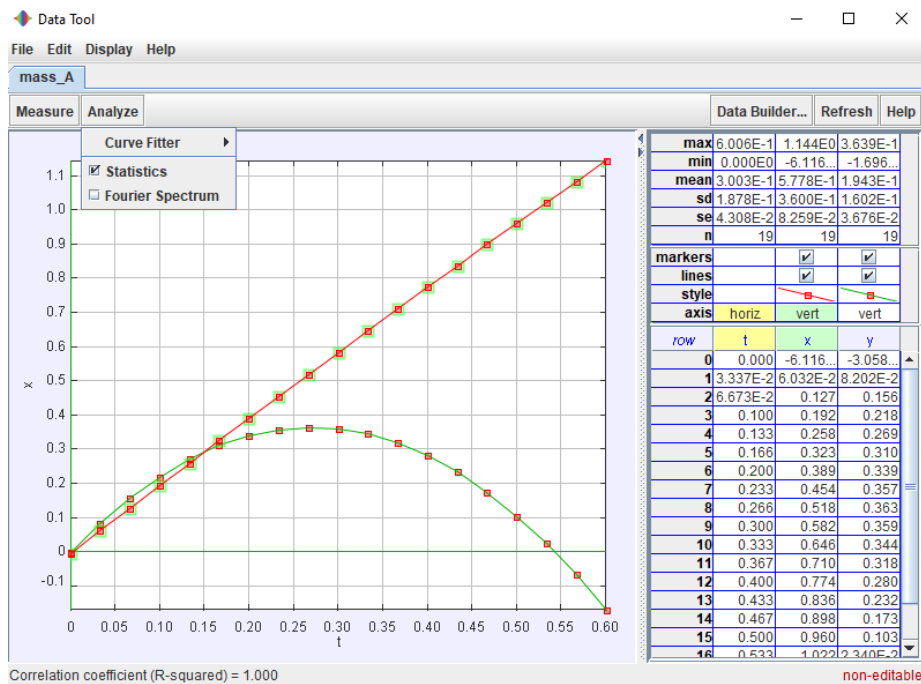
Figur 3.28: Vinduet til **Data Tool**.

**Curve Fitter** lar deg tilpasse datasettet til forhåndsdefinerte funksjoner:

- **Line** lar deg tilpasse etter en linje på formen  $y = at + b$
- **Parabola** tilpasser etter et kvadratisk uttrykk på formen  $y = at^2 + bt + c$
- **Cubic** tilpasser etter et kubisk uttrykk på formen  $y = at^3 + bt^2 + ct + d$
- **Exponential** tilpasser etter et eksponentialuttrykk på formen  $y = a \exp(b(t - c))$
- **Sinusoid** tilpasser etter et harmonisk uttrykk på formen  $y = a \sin(bt + c)$

Resultatet av kurvetilpassningen vises nederst med  $R^2$  som forklarer hvor mye av variansen som forklares av regresjonen.

**Statistics** gir deg tilgang til minimum, maximum, gjennomsnitt, standardavvik, varians, og antall punkter til variablene, se figur 3.29.



Figur 3.29: Statistics for Curve Fitter.

### 3.6 Trackerfiler (.trk, .trz)

Foruten tidligere nevnte filtyper har Tracker to egne filer. Den ene er **Tracker Tab Files** (.trk), som lagrer informasjon om hver tab (videoklipp, koordinat-system, Tracks og Views). Når en slik fil åpnes forblir vinduet slik det var da brukeren lagret, men det lagrer ikke selve videofilene.

Den andre er **Tracker project files** (.trz), som også lagrer videofilene med eventuelle .pdf- eller .html-filer i dokumentasjonen for Tracker.



# Kapittel 4

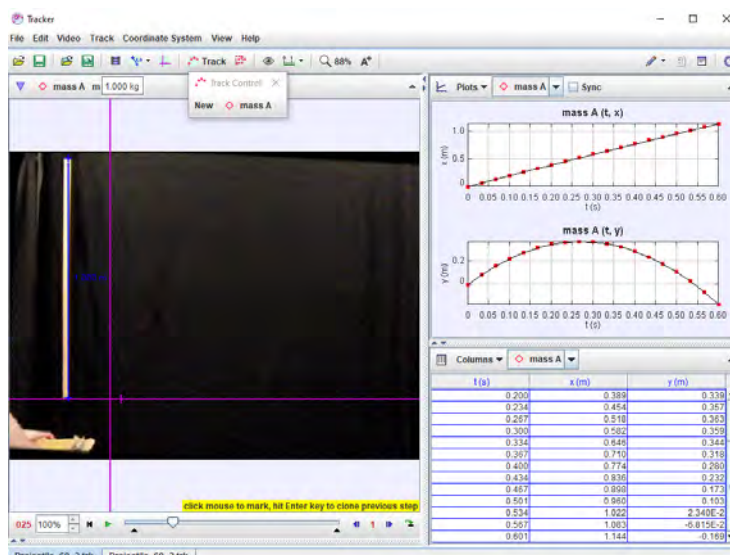
## Videomodellering

I tillegg til videoanalyse er det mulig å lage modeller i Tracker. Det er to typer modeller som kan brukes, en analytisk modell gjennom en funksjon av posisjon, og en dynamisk modell hvor man lager funksjoner for kreftene som virker på et objekt. Den dynamiske modellen er spesielt egnet i forsøk med luftmotstand.

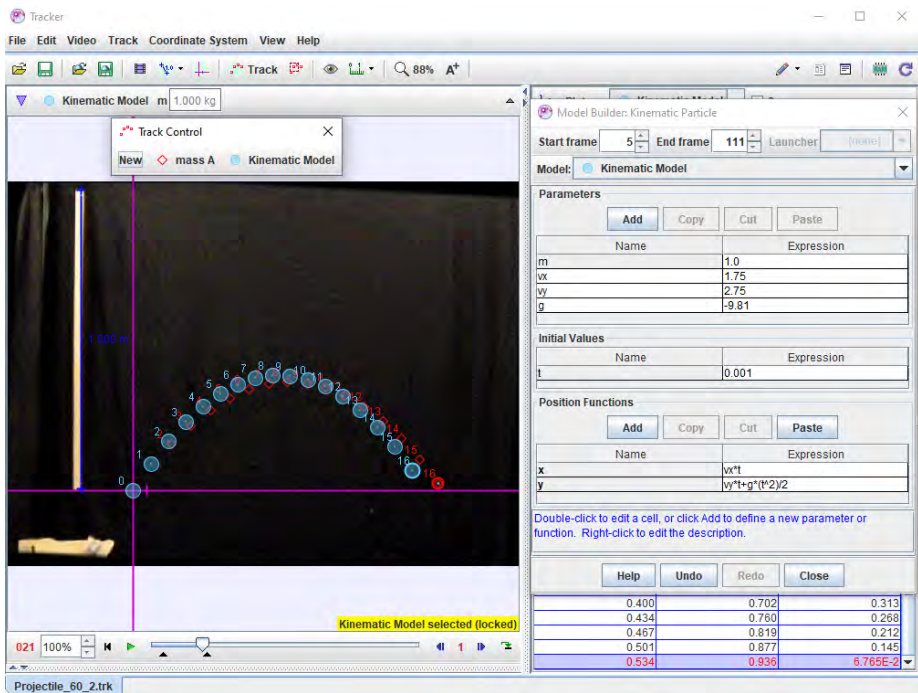
### 4.1 Kast med ball

Et vanlig forsøk er kast med ball. Her kjenner vi til hvilke funksjoner som kan anvendes for å beskrive bevegelsen, og det er derfor en passende introduksjon til modellering.

Når man har en video kan man analysere den på vanlig måte og få et resultat, eller så setter man opp en egen modell. Det kan være lurt å analysere kastet først for å få startverdien til modellen, f.eks gjennom **Autotracker**, se figur 4.1.



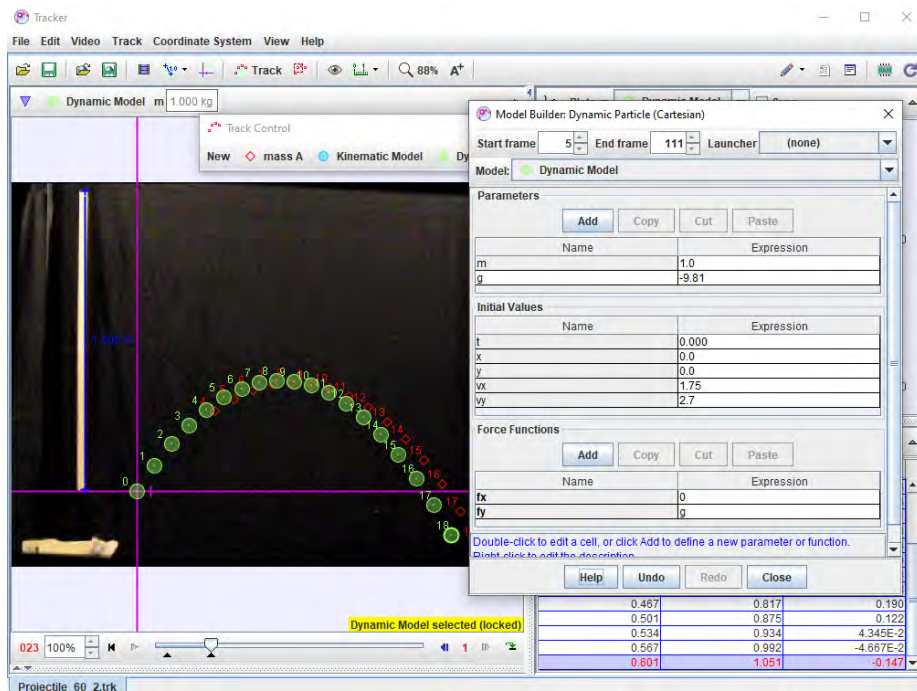
Figur 4.1: Skjerm bilde med markerte steg i videoen. Diagram og tabell viser posisjonene som funksjon av tiden.



Figur 4.2: Figuren viser Model Builder hvor vi har definert verdier for  $v_x$ ,  $v_y$ , og  $g$ . Til venstre ser du både sporet til Autotrackeren (rød), og sporet gjennom modellen beskrevet i Model Builder (blå), dvs.  $x = v_x t$ , og  $y = v_y t + \frac{1}{2} g t^2$

Gå fra **Track** til **Track Control** og velg en ny **Kinematic Model**. Posisjonsfunksjonene lages ved hjelp av de definerte parametrene og tidsvariabelen. Modellen som skapes blir direkte synlig i videovinduet, se figur 4.2.

Den dynamiske modellen arbeider med krefter som virker på objektet, dvs. akselerasjonen som gjør at neste posisjonen i neste bildet kan beregnes. Det er i denne modellen at eksempelvis luftmotstand kan inkluderes som en ekstern kraft. Gå fra **Track** til **Track Control** og velg en ny **Dynamic Model**. I dette eksempelet er det mest hensiktsmessig å bruke kartesiske koordinater. Man kan nå vise modellene samtidig med ballens bane, se figur 4.3. Dette gir oss mulighet til å direkte sammenligne hvor godt modellen passer med den valgte modellen.



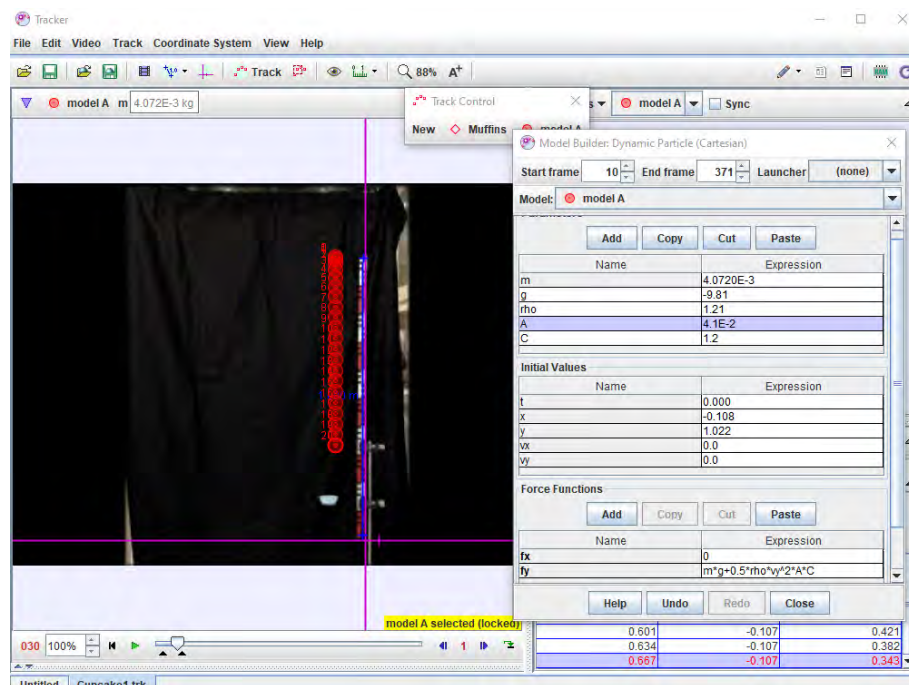
Figur 4.3: Figuren viser Model Builder med tilhørende bane. Her har vi en startposisjon og bruker tid og tyngdekraften til å beregne ballens akselerasjon.



## 4.2 Luftmotstand

Det er ofte nødvendig å ta hensyn til luftmotstand for å få resultatene nøyaktige nok. I dette eksempelet ser vi på hele bevegelsen til en muffinsform. Her skaper vi en dynamisk modell og modellerer hele bevegelsen. Vi ser på to modeller for luftmotstanden, en lineær på formen  $F = -Bv$ , eller en kvadratisk på formen  $F = -\frac{1}{2}C\rho Av^2$ , der  $C$  er en formfaktor,  $A$  er tverrsnittarealet og  $\rho$  er luftens tetthet,  $B$  er en konstant som inneholder disse parametrene. Det er først og fremst arean og formfaktoren  $C$  eller  $B$  som kan varieres, det vil si formen. Den variabelen som er enklest å endre i eksperimentet er massen, som kan gjøres ved å f. eks legge på en binders, eller legge flere muffinsformer oppå hverandre.

I **Model Builder** lager vi modellene som skal brukes. I figur 4.4 ser vi den kvadratiske modellen.



Figur 4.4: Vi begynner med den kvadratiske modellen, og legger merke til at sporet passer ganske godt til bevegelsen i videoen. Her er det mulig å prøve seg frem med ulike konstanter til man oppnår den nøyaktigheten man ønsker.



Nå ser vi på bevegelsen til flere muffinsformer med ulik vekt. Her slippes 8 muffinsformer samtidig og man ser tydelig at de letteste påvirkes mest. Her ser vi på tre former med ulike masser ( $m = 1.4$  og  $7$  binders) samtidig som vi skaper to modeller for hver form. En med den lineære modellen og en med den kvadratiske. I figur 4.5 ser vi resultatene. Man kan da se direkte hvor godt modellen fungerer. Den lineære modellen fungerer relativt bra for de tyngre muffinsformene, men gir helt feil resultat for de letteste. Parameteren  $B$  har samme verdi i alle tilfellene. Det er mulig å få den lineære modellen til å stemme for alle observasjoner, men man må da endre på en konstantparameter. Den kvadratiske modellene fungerer for alle massene, som er synlig på bildet. Det er viktig å poengtere at man ikke trenger å se i tabeller eller diagram for å se hvor bra modellen er – det er synlig i videoen, som gjør det mer virkelighetsnært for elever. Dermed blir den hverdagslige situasjonen knyttet direkte til modellen, og man slipper å veksle mellom ulike representasjonsformer.



Figur 4.5: Skjerm bilde av formens eksperimentelt bestemte posisjoner og posisjon gitt fra modellen. Den lineære modellen markeres som en sirkel, den kvadratiske med en diamant og muffinformens posisjon med et triangel.



## Kapittel 5

# Gravitasjonskonstanten på månen

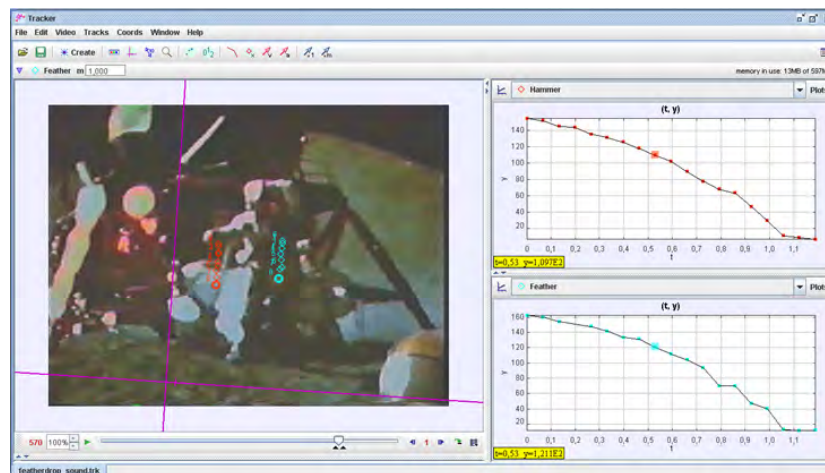
*Dette avsnittet er skrevet av J.E. Hagen <sup>1</sup> og Jonas Persson.*

Å bestemme akselerasjonen for objekter i fritt fall er et standard eksperiment i fysikk, som alle elever på et eller annet tidspunkt utfører. Det finnes mange metoder for å gjøre dette, avhengig av målsetning og tilgjengelig utstyr i klasserommet. Utviklingen av videoanalyseverktøy har gitt oss enda en metode. Bruk av en video gjør det mulig å studere mer komplekse og eksotiske situasjoner, eller hverdagssituasjoner der elevene selv har gjort et opptak med for eksempel et mobiltelefonkamera. Hvis en video er tilgjengelig og målestokken er kjent, så er det mulig å analysere bevegelsen. En av de mest spektakulære demonstrasjonene av objekter i fritt fall ble utført på månen av kommandør for Apollo 15-romferden, David Scott. Han holdt en hammer og en fjær ut fra kroppen, slapp dem samtidig, og fordi de ble sluppet i et vakuum falt de med samme hastighet slik Galileo Galilei konkluderte 400 år tidligere. Videoen, som er tilgjengelig på NASAs hjemmeside [1], gjør det mulig å bestemme akselerasjonen grunnet gravitasjonen på månen ved hjelp av videoanalyse.

Uheldigvis er kvaliteten på videoen ganske dårlig. Det er vanskelig å skille hammeren og fjæren fra bakgrunnen, det finnes ingen kalibrert målestokk, og kameraet er plassert litt skjevt. Både hammeren og fjæra er uskarpe, men det er mulig å observere dem i fritt fall i mer enn 10 bilder på videoen, spesielt i første del av fallet. Det er derfor mulig å skaffe nok informasjon til å uttrykke posisjonen som en funksjon av tiden. Hvor skjevt kameraet er plassert kan bestemmes ved å bruke posisjonen til hammeren rett etter at den er blitt sluppet og posisjonen hvor den lander. Disse punktene gjør det mulig å bestemme den sanne  $y$ -aksen, som er omtrent 4 grader fra  $y$ -aksen til kameraet. Målestokken er mer usikker, men informasjon tilgjengelig fra nettstedet sier at fallet var omtrent 1,6 m. Dette stemmer overens med den antatte høyden til astronauten, omtrent 2,0 m med hjelm, men er allikevel en kilde til usikkerhet. Ved hjelp av videoanalyseprogrammet Tracker, ble posisjonene til både hammeren og fjæra avmerket og registrert. Den nøyaktige posisjonen i hvert bilde kan være vanskelig å finne, og resultatet vil derfor variere noe. For å unngå bias, bør plottvinduet ikke være synlig når posisjonene blir markert.

---

<sup>1</sup>Vardø videregående skole, Kaigata 6, 9950, VARDØ



Figur 5.1: Videovinduet og plotter av  $y$ -posisjonen til hammeren og fjæra i Tracker. Den korrekte  $y$ -aksen er vist i videovinduet

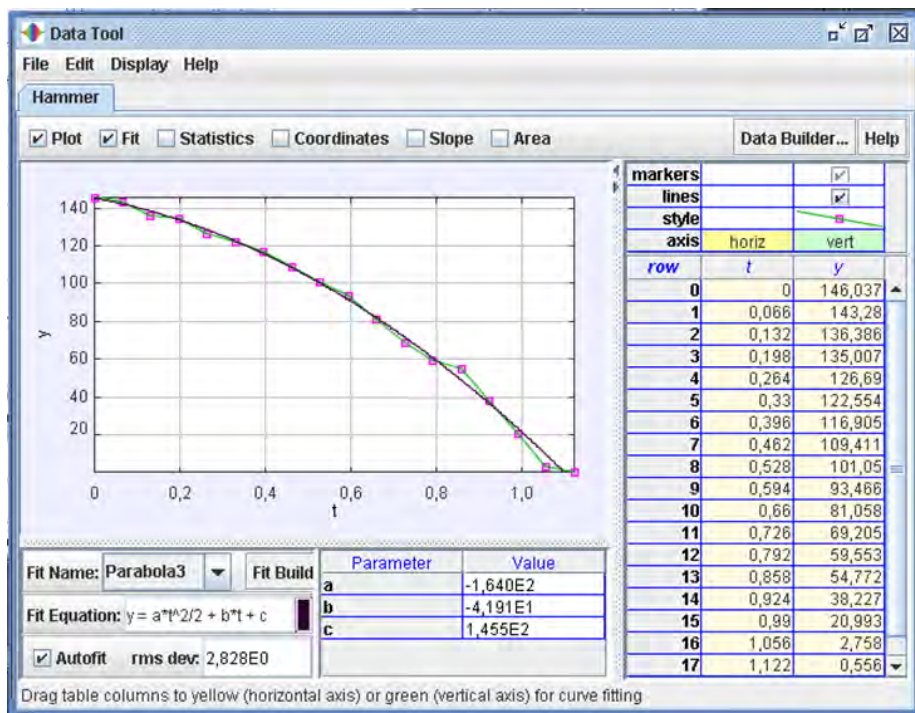
Det er nyttig å bla fram og tilbake mellom bildene for å skille hammeren og fjæra fra bakgrunnen. Fra posisjonsdataene får man fram en tilpasset kurve, med en andregradsligning som beskriver fallbevegelsen. Ved å identifisere akselerasjonen i ligningen kan den bestemmes. Bevegelsen for en gjenstand i fritt fall kan beskrives av bevegelsesligningen:

$$y = y_0 + v_y t + \frac{1}{2} a t^2$$

Der  $y_0$  er høyden objektet slippes fra,  $v_y$  farten den har i startøyeblikket og  $a$  er akselerasjonen. Ettersom vi ikke starter tiden på eksakt null får vi en startfart. Etter å ha markert fallbevegelsen til hammeren fikk vi denne bevegelsesligningen: 5.1.

$$y = 1.455 - 0,42 \cdot t - 1.64 \frac{t^2}{2}$$

Med andre ord en akselerasjon på  $1.6 \text{ m/s}^2$  nedover.



Figur 5.2: Anpassning av bevegelsekurva till en ekvation som ger akselerasjonen (skalan gis i cm)

Resultatet fra Anpassning vises i figur 5.2. Vi fikk akselerasjonen  $1,6 (\pm 0,3)$  for fjæra og  $1,6 (\pm 0,4) \text{ m/s}^2$  for hammeren ved å gjenta avmerkingen av bevegelsen flere ganger. Disse verdiene stemmer godt overens med den beregnede verdien  $1,6 \text{ m/s}^2$  og er tydelig forskjellig fra verdien på jorden. Grunnet vanskelighetene med å markere korrekte posisjoner kan det forventes verdier mellom  $1,3 - 2,2 \text{ m/s}^2$  for elever og andra som vil gjenta forsøket. Disse resultatene burde overbevise dem om at gravitasjonen er mindre på månen og at den stemmer overens med deres egne beregninger med Newtons gravitasjonslov. Resultatet er en sterk indikasjon på at månelandingen virkelig fant sted. Det er mulig å simulere lavere gravitasjon ved å redusere bildehastigheten på videoen. Ved å spille av Apollo 15-videoen raskere, vil de fallende objektene få samme akselerasjon som på jorda, men bevegelsene til astronautene vil bli unaturlige for en person i en romdrakt. Bruk av videoanalyse er et billig alternativ for visse kinematikk eksperimenter, og alt som trengs er et standard webkamera for å registrere eksperimentet. Fordi mange videoer, som Apollo 15-videoen, er tilgjengelig på nettet, har man dessuten mulighet til å studere et rekke eksotiske situasjoner.

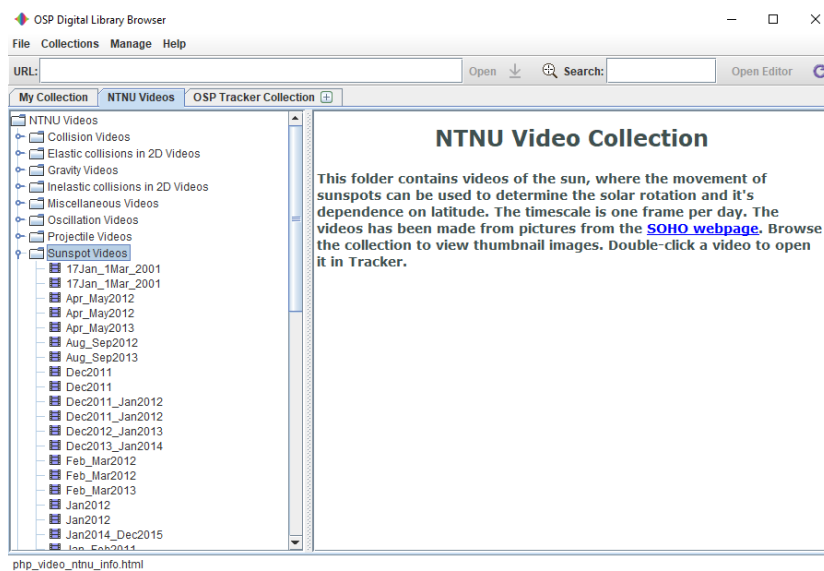


## Kapittel 6

# Solflekker – Solens rotasjon

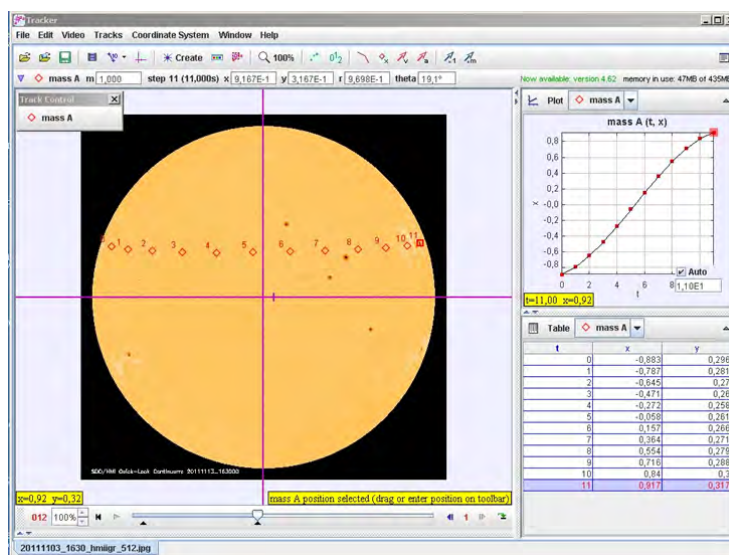
At solen har sine flekker og at vi kan observere dem ved hjelp av teleskop eller bilder fra Internett vet mange. Men det er faktisk mulig å få ut mer informasjon enn dette. Ved å studere solflekker kan man vise at solen roterer og at rotasjonen er differensiell, dvs. at den roterer med ulik hastighet avhengig av hvilken breddegrad man ser på (man bruker de samme begrepene om lengde- og breddegrad som på jorden). Solflekker oppstår i nærheten av ekvator, typisk mellom  $30^{\circ}\text{N}$  og  $30^{\circ}\text{S}$ . Deretter kan man studere en solflekk og se hvor den forflytter seg og dermed bestemme solens rotasjon for den breddegraden.

Det finnes et stort antall observasjoner og satellitter dedikert til å studere solen. En god ressurs for dette er bilder fra *SOHO*-satellitten, som ligger i Lagrangepunktet L1 [14], mellom Jorden og Solen. Disse bildene kan lastes ned gjennom **OSP Digital Media Browser**, se 6.1.



Figur 6.1: Bildet viser arkivet av videoer hentet fra **SOHO**-nettsiden[12].

Disse filmene består av ett bilde per døgn og omfatter 1–2 måneder. Når man laster inn bildene i **Tracker** gjennomgår man den vanlige prosedyren med lengdeskala, koordinatsystem og tidsskala (jf. seksjon 3.4). Koordinatsystemet velges slik at origo havner i sentrum av solen. Lengdeskalaen velges slik at solens radius får lengden 1, dvs. solens diameter skal være 2. Tidsskalaen viser antall bilder per døgn. Har du ett bilde per døgn velger du skalaen 1 sekund (= 1 sekund mellom hvert bilde), men husk at sekund da er et døgn. Om du har 16 bilder per døgn setter du tidsskalaen til 1/16. I **Tracker** markerer du den/de solflekkene du vil studere på vanlig måte, se figur 6.2. Du får nå  $x$ - og  $y$ -koordinatene i tabeller og/eller grafer.



Figur 6.2: **Tracker** med en solflekk markert.

For å bestemme lengdegraden på solflekken bruker du  $y$ -koordinaten:

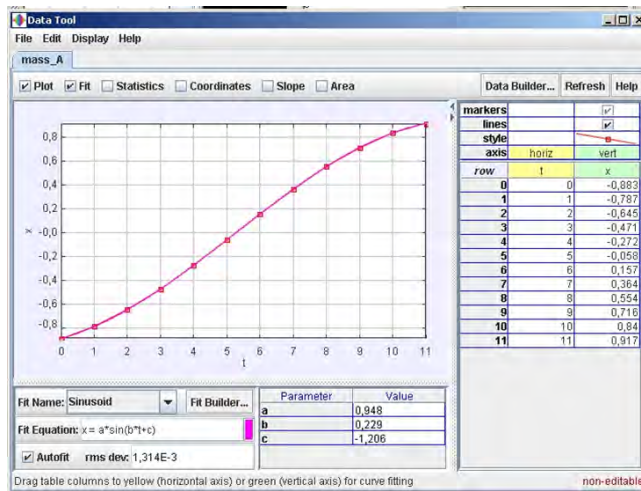
$$\sin \varphi = \frac{y}{R}$$

Her kan du nå studere om lengdegraden endres eller om den holder seg konstant. For å få lengdegraden i tabellen kan man bruke **Databuilder**.

$x$ -koordinaten kommer til å variere med tiden og det skal vi anvende for å bestemme rotasjonen. Studerer man grafen av  $x$ -posisjonen, så ser du en sinuslignende funksjon. Gå til **Analyze**. og anvend **Sinusoidal** kurvetilpassning.  $x$ -posisjonen kommer da til å tilpasse ligningen  $x = a \sin(b \cdot t + c)$ , der parameteren  $b$  tilsvarer vinkelfrekvensen,  $\omega = \frac{2\pi}{T}$ . Her kan det bli et problem med tilpasningen, som gjør at det anbefales å selv legge inn startverdien på  $a$  og  $b$  som 1 og 0.25 respektivt, og deretter trykke på **refresh** eller markere **autofit**. Observer at parameteren  $a$  skal ha en verdi på ca. 1, store verdier viser at tilpasningen har funnet feil minimum. Man kan også selv variere verdiene for å finne en bedre tilpassning.

Gjennom å ta parameteren  $b$  kan du bestemme rotasjonstiden, men observer her at perioden gis i antall dager.





Figur 6.3: Tilpassning til solflekkens posisjon med en sinusfunksjon.

Ved å studere solflekker på ulike lengdegrader kan man se at solen har en differensiell rotasjon. Solens rotasjon er ved ekvator 25.05 dager og ved 16° lengdegrad, 25.38 dager, dette er den sideriske rotasjonen, dvs. i forhold til stjernene. Den man får fra *SOHO*-data er den synodiske (observert fra jorda) som må korrigeres for å finne den sideriske (relativt stjernene) rotasjonen.

$$\frac{1}{T_{\text{siderisk}}} = \frac{1}{T_{\text{synodisk}}} + \frac{1}{T_{\text{jord}}},$$

der  $T_{\text{jord}}$  er tiden det tar for jorden å gå en runde rundt solen.

I vårt eksempel får vi solens rotasjonstid til 25.52 dager, ved lengdegraden, ca. 16°.



## Kapittel 7

# Coefficient of restitution

I sportssammenheng oppfører baller seg som vi forventer, deterministisk. Dette innebærer blant annet at de må sprette på en kontrollert måte. Når en ball spretter, mister den noe av den kinetiske energien i kontakt med bakken, dette gjøres ved å deformere ballen og varme den opp. Vi har et uelastisk støt.

Om vi antar at spretten foregår mot en stasjonær gjenstand (et gulv), vil vi være i stand til å beregne energitapet ved å se på hastigheten til ballen før og etter spretten.

Gjennopprettingskoeffisienten (COR), er definert som forholdet mellom hastigheten før og etter støtet:

$$C_R = \frac{v_{etter}}{v_{før}}$$

Vi kan også definere den utifra hvor man slipper ballen og hvor høyt den spretter:

$$C_R = \sqrt{\frac{h_{etter}}{h_{før}}}$$

Coefficient of restitution anvendes ofte i golfsammenheng der man ønsker at den skal være så høy som mulig. Man har da satt en øvre grense for COR til 0.83, der en typisk verdi ligger på rundt 0.78.

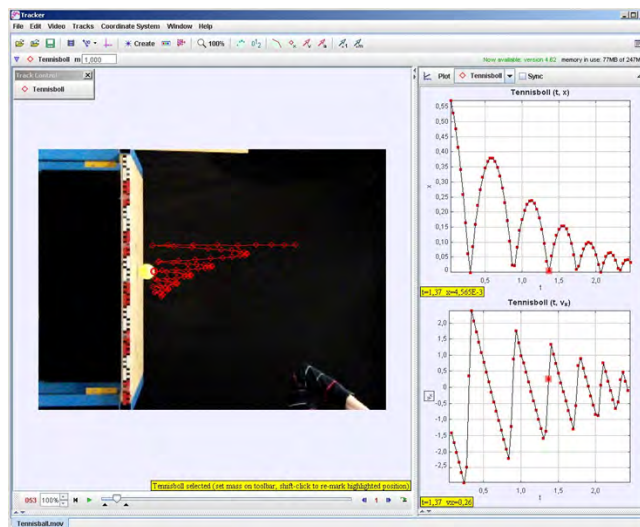
Det internasjonale bordtennisforbundet har bestemt at en bordtennisball som slippes fra 30.5 cm på en standard stålblokk skal sprette opp 24–26 cm, hvilket tilsvarer en COR på 0.89–0.92.

Det er relativt enkelt å gjennomføre eksperimentet selv, enten med video, eller med å spille inn lyden av spretten og analysere tidsintervallet. I dette tilfellet skal vi vise hvordan dette gjøres med en video.

En ball slippes fra en høyde og man filmer mens ballen gjennomgår et visst antall sprett. Filmen kan så lastes inn i **Tracker** og analyseres.

I figur 7.1 vises en tennisball som slippes fra en viss høyde. Da dette er en relativ måling betyr ikke den absolutte høyden noe. Selv om vi slipper ballen i  $y$ -retning er det like greit å analysere et horisontalt bilde.

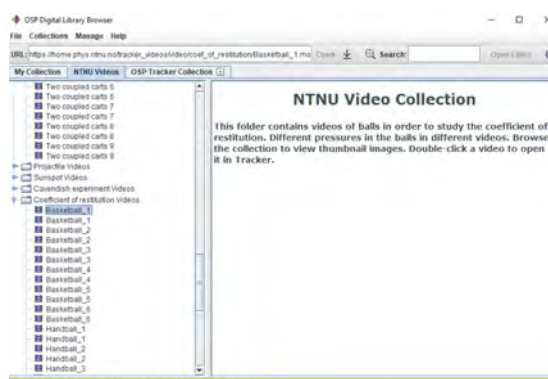
Når man skal analysere COR fra høyden på sprettene, må man passe på å legge koordinatsystemet slik at det laveste punktet ligger så nære null så mulig. Her er det viktig at markeringspunktet blir nøyaktig. For kontrastens skyld bør



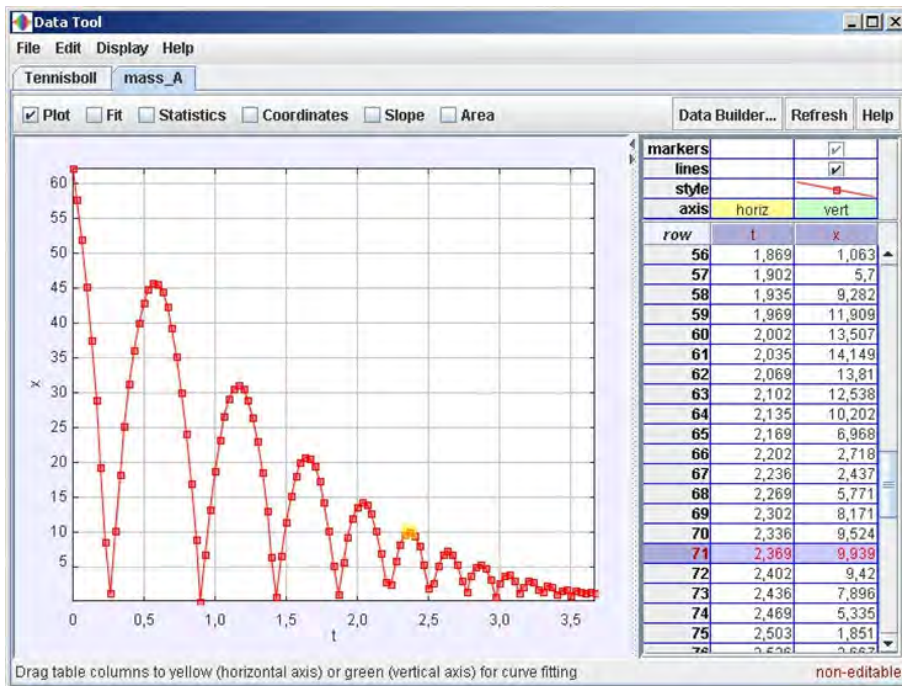
Figur 7.1: Analyse av en sprettende tennisball

den øvre delen av ballen brukes. I dette tilfellet kommer man til å få en stor akselerasjonsforandring som gjør at **Autotracker** kan få et problem med at objektet havner utenfor søkeområdet. For å unngå dette kan man øke størrelsen til søkeområde, eller bruke **Search Next**. Denne funksjonen gjør at man kan flytte søkeområdet om **Autotracker** ikke finner rett og søker igjen bilderutene med **Search This**

Fra figur 7.1 ser vi at vi kan bruke både fartsgrafene og posisjonsgrafene til å bestemme COR. Det er viktig at man ser på flere spretter og forsøker å få en så bra verdi som mulig. I forsøket over fikk jeg en verdi på COR på 0.77. Her kan man gjennomføre forsøket med baller fra ulike idretter og se hvilke COR de har på ulike underlag. Man kan også utvikle forsøket videre og studere hvordan lufttrykket i ulike baller påvirker spretten og derfor COR. Det finnes enkle målere som man kan bruke til å måle trykket i en ball. Et stort antall videoer med ulike baller og ulike lufttrykk er tilgjengelige gjennom NTNU sitt videobibliotek delt i **Tracker**, se figur 7.2.



Figur 7.2: Bildet viser et utvalg av videoer av ulike baller som spretter.



Figur 7.3: Resultatet av håndballen som slippes.

I et av forsøkene bruker jeg en håndball der jeg varierte trykket fra 0.6 kg/cm<sup>2</sup> til 0.3 kg/cm<sup>2</sup>.

I figur 7.3 ser man hvor høyt ballen spretter og dette kan brukes til å beregne COR. Ved å studere sammenhengen mellom trykk og COR ser man tydelig at en ball med høyt trykk spretter bedre.

Trykk (kg/cm <sup>2</sup> )	COR
0.60	0,827
0.45	0,801
0.30	0,783

Her kan man prøve ulike baller og underlag for å se hva som er best. I slike undersøkelser basert på relativt enkle fysiske sammenhenger kan man avdekke forhold som kan være viktige i utvikling av utstyr i idretter.

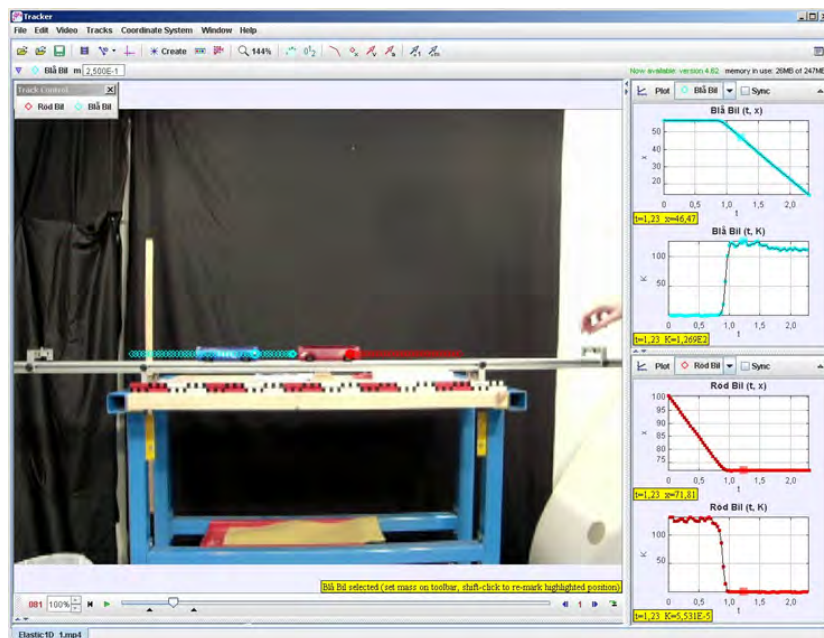


## Kapittel 8

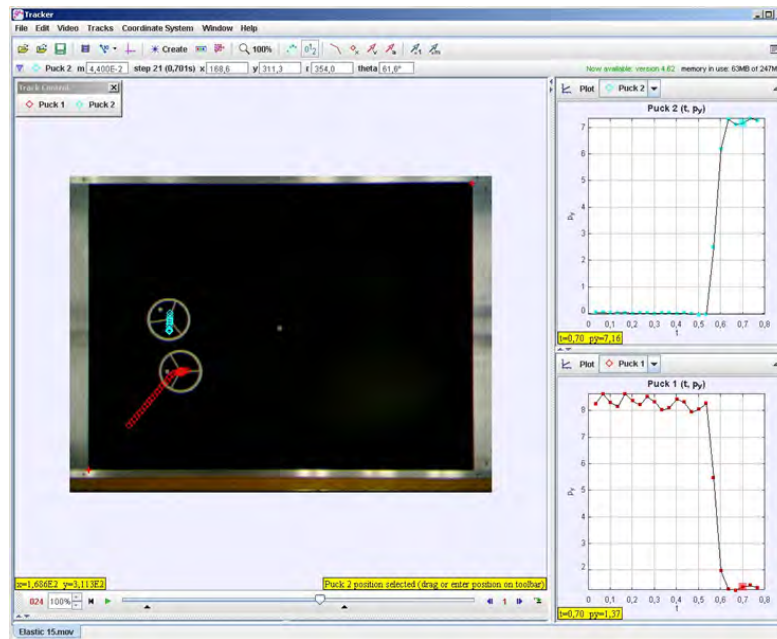
# Kollisjoner

Eksperimenter der videoanalyse er spesielt egnet, er kollisjoner med to eller flere objekter samtidig. Vi har begrenset oss til målinger i to dimensjoner. Det enkleste kollisjonsforsøket kan man gjøre mellom to vogner på en bane med lav friksjon. Prosedyren med **Tracker** er å spille inn en video og laste den inn. Kalibrering og markering på samme måte som tidligere, se underseksjon 3.4.3.

Her er det viktig at man har en unik markering på hver vogn for å gjøre markeringen lettere for **Autotracker**. Man kan få store endringer i bevegelsen som gjør at objektet havner utenfor søkeområdet, så vær varsom ved bruk av **Autotracker**. Når markeringen er gjort kan man vise posisjonen som funksjon av tid, men også bevegelsesmengden og kinetisk energi dersom massen er spesifisert. Det vil si at man får en direkte sammenligning av disse før og etter kollisjonen. Dette vises i figur 8.1.

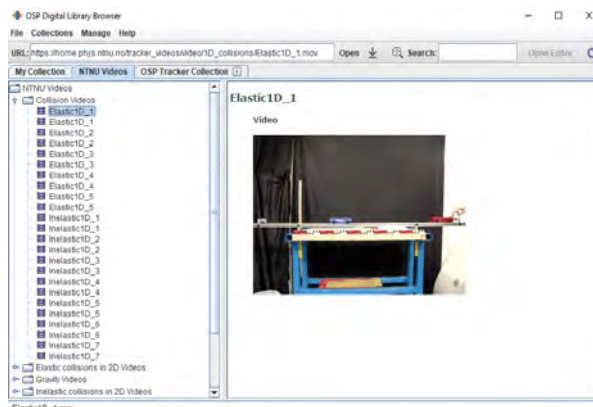


Figur 8.1: Kollisjon i én dimensjon. Energi og posisjon vises i grafen.



Figur 8.2: Kollisjon i to dimensjoner. Grafer med bevegelsesmengde i  $y$ -retning.

Siden vi ikke har en friksjonsfri bane, vil det finnes små ujevnheter som påvirker bevegelsen. Dette blir spesielt tydelig i høye hastigheter, og gjør at den kinetiske energien ikke vil være bevart i kollisjonen. Her er den kinetiske energien før kollisjonen er 1.27 J (lengdeskalaen er i cm) og 1.17 J etter. Vi kan også gjøre eksperimenter i to dimensjoner, om vi f. eks har et *Air Hockey*-bord. Her kommer man da virkelig til å utnytte mulighetene med videoanalyse. Med massene spesifisert kan vi bestemme bevegelsesmengden og den kinetiske energien i alle komponentene. Dette gir en unik mulighet til å virkelig se at bevegelsesmengden bevares i hver komponent. Det finnes et utvalg videoer av kollisjoner på NTNU sin kolleksjon i **OSP Digital Library Browser**, se figur 8.3



Figur 8.3: Videoer av kollisjoner. Det er et utvalg av elastiske og uelastiske støt i både én og to dimensjoner.



# Bibliografi

- [1] Logger Pro<sup>®</sup> 3, hentet 22. juli 2021  
<https://www.vernier.com/product/logger-pro-3/>
- [2] Motion Tracking & Analysis Software WINalyze, hentet 22. juli 2021  
<https://winalyze.com/>
- [3] ProAnalyst Motion Analysis Software, hentet 22. juli 2021  
<https://www.xcitex.com/proanalyst-motion-analysis-software.php>
- [4] PASCO Capstone Software, hentet 22. juli 2021  
<https://www.pasco.com/products/software/capstone>
- [5] CameraSim Web App, hentet 1. mars 2021  
<https://camerasim.com/camerasim-free-web-app/>
- [6] Tracker Video Analysis and Modeling Tool, hentet 9. juni 2021  
<https://tracker.physlets.org/>
- [7] Tracker Github Repository, hentet 29. mai 2021  
<https://github.com/OpenSourcePhysics/tracker>
- [8] Tracker OSP Tools, hentet 29. mai 2021  
<https://www.compadre.org/osp/webdocs/Tools.cfm?t=Tracker>
- [9] Tracker Installer Help, hentet 29. mai 2021  
[https://tracker.physlets.org/installers/installer\\_help.html](https://tracker.physlets.org/installers/installer_help.html)
- [10] Tracker Help, hentet 12. juni 2021  
<https://physlets.org/tracker/help/>
- [11] The Apollo 15 Hammer-Feather Drop, Hentet 30. juni 2021  
[https://nssdc.gsfc.nasa.gov/planetary/lunar/apollo\\_15\\_feather\\_drop.html](https://nssdc.gsfc.nasa.gov/planetary/lunar/apollo_15_feather_drop.html)
- [12] SOHO Data, Hentet 30. juni 2021  
<https://soho.nascom.nasa.gov/data/data.html>
- [13] Tracker Online, hentet 30. juni 2021  
<https://tracker.physlets.org/trackerJS/>
- [14] Lagrange-punkter, hentet 30. juni 2021  
<https://snl.no/Lagrange-punkter>

ISBN 978-82-8054-205-2

