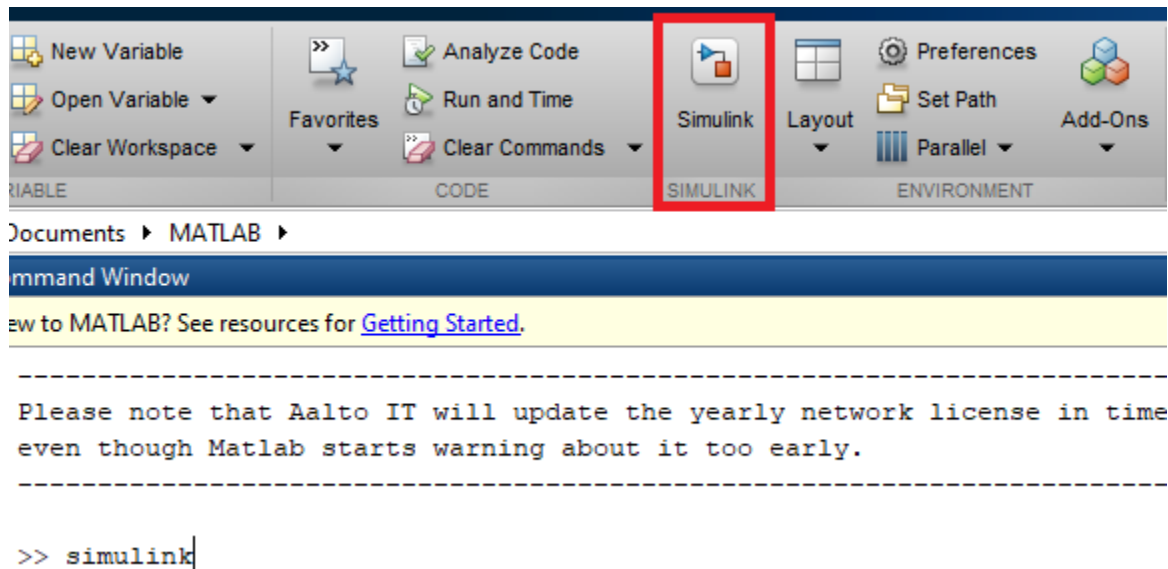


# Tilakone

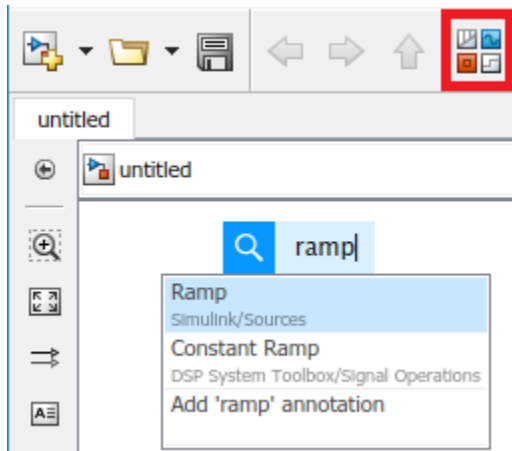
*Arvostelu: 5p kun olet saanut samanlaiset tulokset kuin tämän dokumentin kuvissa.*

Tämän tehtävän tarkoitus on oppia käyttämään Simulink tilakoneita mahdollisimman lyhyessä ajassa. Tässä esitellään olennaiset tekniikat, joita tarvitaan seuraavassa harjoituksessa. Avaa Matlab ja sitten Simulink tästä Ikonista tai kirjoittamalla Command Windowiin "simulink":



*Kuva 1: simulinkin avaus*

Sitten avaa library browser (alla olevassa kuvassa neliöity ikoni) ja lisää malliin 'Ramp' sources kirjastosta ja 'Scope' sinks kirjastosta. uudemmissa Simulink versioissa voit tuplaklikata valkoista taustaa ja käyttää hakutoimintoa.



Kuva 2: Rampin asettaminen työtilaan

Kytke "ramp" sekä "scope" toisiinsa ja aja malli Play ikonia klikkaamalla. Scopeen pitäisi tulla ramppi. Sitten lisää Stateflow kirjastosta 'Chart'. Tehdään tästä chartista tilakone, joka valitsee ohjaussignaaleja joystickin ja napin välillä. Tuplaklikkaa chart auki ja avaa Model Explorer tästä ikonista:



Kuva 3: Model explorer

Lisätään inputit joystickille ja napille. Valitse 'Add' valikosta 'Data'. Ilmestyy rivi. Vaihda 'scope' kohtaan 'input' ja nimeksi 'Joystick'. Oikeassa reunassa laita typeksi double. Sitten luo toinen samanlainen input nimellä 'button'. Sitten luo output nimellä 'ControlSignal'.

Tehdään aluksi joystick ohjauslogiikka. Jos joystick input on alle 10, ohjaus on sama kuin joystickin arvo. Jos se on yli 10, niin arvo on:

$$10 + (\text{joystick} - 10) * 0.1$$

Luo Matlab funktio ja anna joku nimi esim. JoystickControl1:

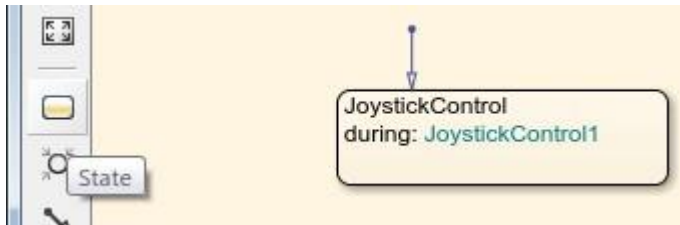


Kuva 4: matlab-funktio

Tuplaklikkaa funktiota ja kirjoita seuraava Matlab koodi:

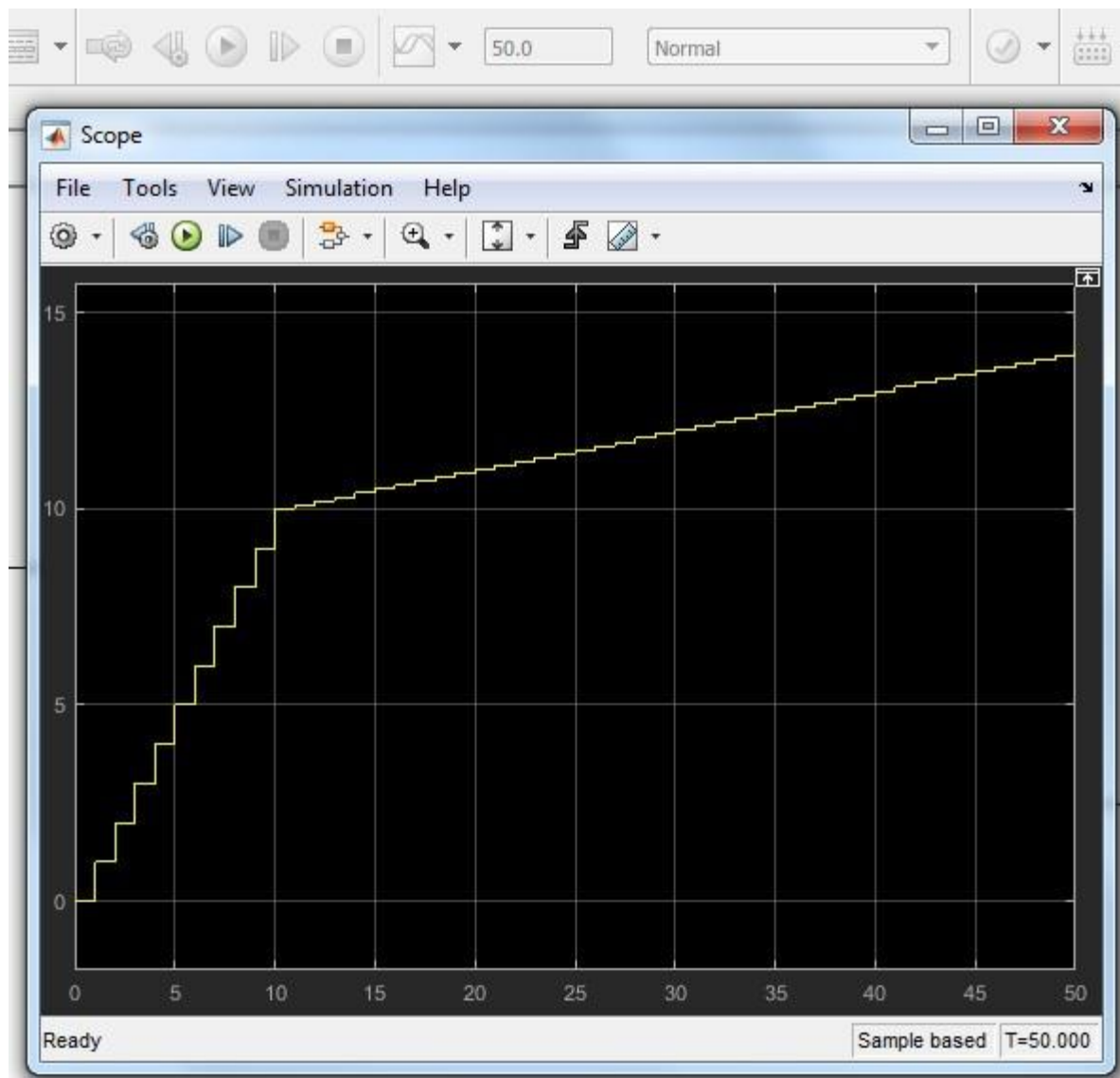
```
function JoystickControl1
    if joystick
    < 10
        ControlSignal = joystick
    else
        ControlSignal = 10 + (joystick-10) * 0.1
    end
```

Sitten luodaan tila (State) ja kutsutaan tätä funktiota (klikkaa state ja raahaa state keltaiselle alueelle):



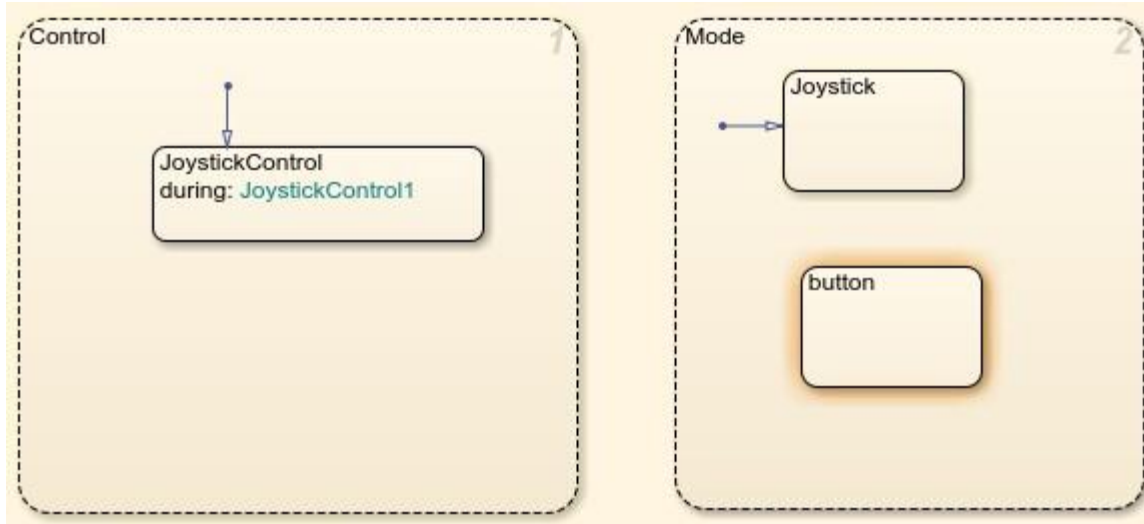
*Kuva 5: Tilan luominen*

'during:' kohtaan kirjoitettu koodi suoritetaan syklisesti (kuten PLC) niin kauan kuin ohjelman suoritus on tässä tilassa. Tilakone on samankaltainen kuin PLC ohjelmoinnin SFC, jossa myös siirrytään transitioiden avulla tilasta toiseen. SFC on kuitenkin varsin rajoittunut verrattuna tilakoneeseen, kuten tullaan huomaamaan seuraavassa harjoituksessa, jossa toteutetaan kiehutusvesireaktorin pääkiertopumppujen ohjaus. Lisää simulaation kestoksi 50 (kts kuvan yläosassa oleva kenttä) ja yhdistä ramppi chartin joystick inputtiin ja scope chartin ControlSignal outputtiin. Aja ja avaa scope, pitäisi näyttää seuraavalta:



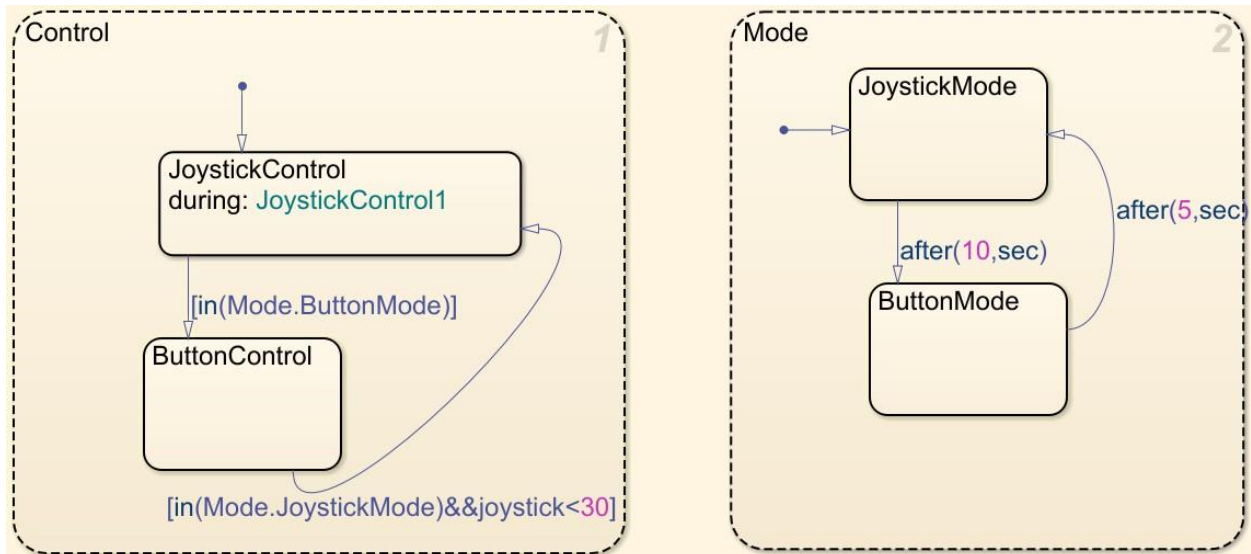
*Kuva 6: Odotettu ulostulo*

Lisää tiloja seuraavan kuvan mukaan. Sitten klikkaa oikealla kaikkien tilojen ulkopuolelle ja valitse Decomposition/Parallel (AND), jolloin tulee katkoviivat, joilla kuvataan rinnakkaiset tilat. Nyt 'Control' ja 'Mode' tilat kummatkin suoritetaan jokaisella syklillä. Numero 1 oikeassa yläkulmassa tarkoittaa että 'Control' suoritetaan ensin. Pieni nuoli (Default transition) pitää lisätä vasemman palkin menusta, jotta tilakone tietää, mihin tilaan mennään, kun ohjelma käynnistyy.



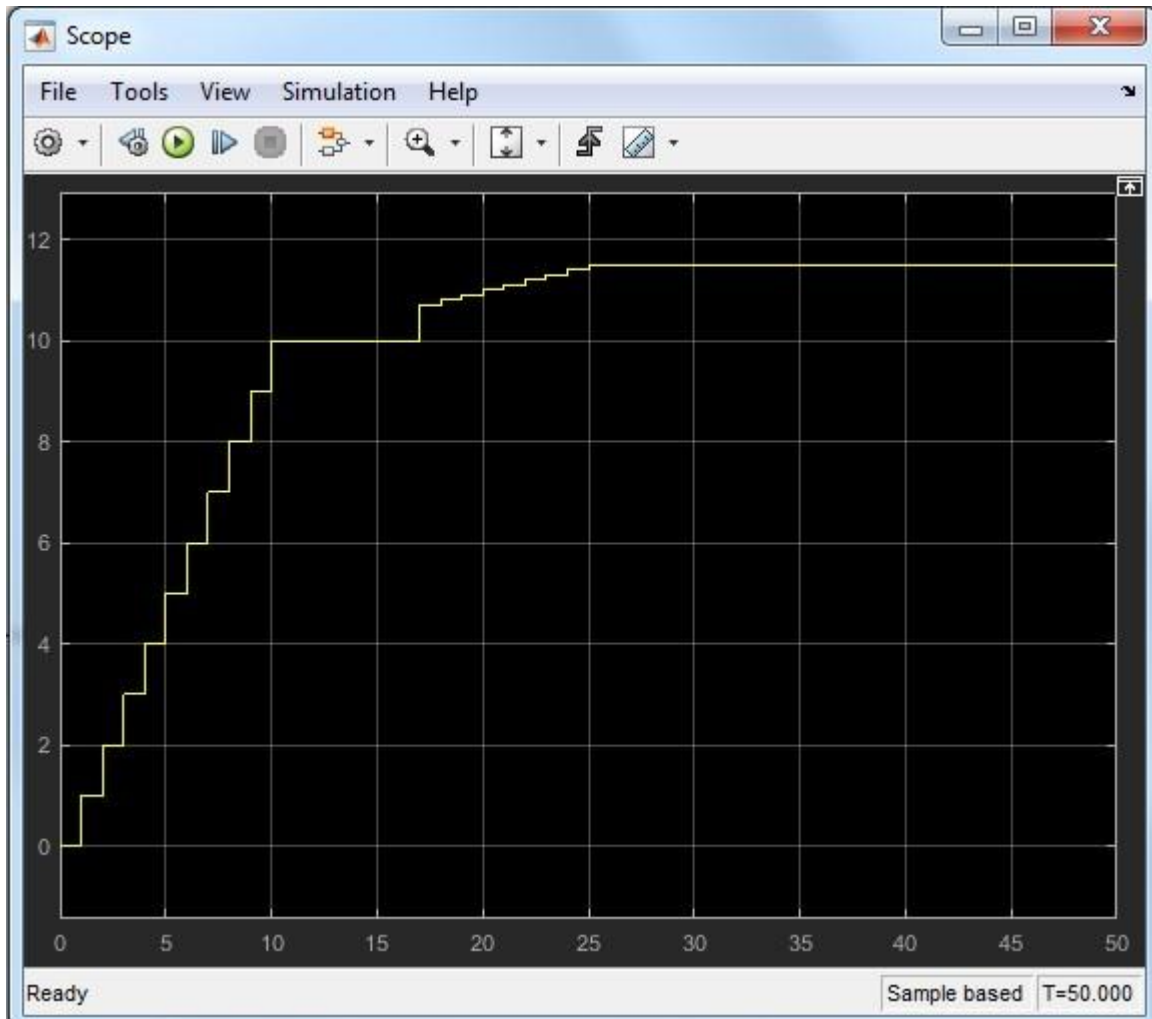
Kuva 7: rinnakkaiset tilat

'Button' niminen tila ja 'button' niminen chart input aiheuttavat ristiriidan. Muuta nimet alla olevan kuvan mukaisesti. Lisää myös tilojen väliset transiitiot ja niiden ehdot. Transition tilojen välille piirretään siten että klikataan tilan reuna vasemmalla ja raahataan toisen tilan reunaan. Sitten klikataan transiitiota ja hakasulkuja ja kirjoitetaan mikä tahansa boolean ehto (sama idea kuin PLC:n SFC:ssä). Tilakoneen transiitiossa voi käyttää myös funktiota `in(<tilan nimi>)` joka palauttaa true jos olla kyseisessä tilassa. `after()` syntaksi on poikkeus – sitä ei kirjoiteta hakasulkujen sisään – transiitio laukeaa kun on oltu kyseisessä tilassa `after()`:ssa määritelty aika. Ehdossa voi käyttää AND (&&) ja OR (||) kunhan ehto voidaan evaluoida true/false.



Kuva 8: Transiitiot

Aja ohjelma. Pitäisi tulla seuraava kuva. Kun ollaan Button modessa, mitään ei tapahdu ja ulostulon arvo ei muutu. Sitten kun joystick arvo on yli 30 (joka on 30s simulaation alkamisesta), ButtonControl tilasta lähtevä transiitio on aina false ja ohjaussignaalin arvo ei muutu.



Kuva 9: Odottetu ulostulo

Tämän harjoituksen tarkoitus oli mahdollisimman nopeasti oppia käyttämään tarpeellisia ominaisuuksia Simulink Stateflowsta. Itse ohjauslogiikka ei ole hyödyllinen ja rinnakkaisten tilojen käyttö tässä nimenomaisessa tapauksessa ei olisi ollut välttämätöntä. Näiden Simulink tekniikoiden avulla voidaan seuraavassa harjoituksessa tehdä realistinen ohjauslogiikka kiehutusvesireaktorin pääkiertopumpuille.