Nodene våre er som følgende:

1. signal\_source.py som gjør jobben med å lese det ufiltrerte signalet fra filen signal\_raw.dat og publiserer denne dataen på signal\_raw topicen på en frekvensen 20Hz
2. lp\_filter.py noden vår abonnerer på signal\_raw topicen og bruker så et low-pass filter på den abonnerte dataen. lp\_filter publiserer også det filtrerte signalet på topicen siginal\_filtered Den bruker da koeffisientene a1, a2, b1 og b2 for å gjøre filtreringen.
3. present\_data.py abonnerer på både signal\_raw og signal\_filtered og logger begge ut til konsollvinduet.

Topics som er brukt i vårt system er som følgende:

1. signal\_raw som er topicen som brukes for å publisere ufiltrert data fra signal\_source.py og blir så abonnert på i lp\_filter.py og present\_data.py
2. signal\_filtered blir brukt til å publisere filtrert data fra lp\_filter.py, og blir så abonnert på av present\_data.py

Servicen vi bruker er definert i filen «do\_filter\_calc.srv» og definerer input variablene a1, a2, b1, b2, ykm1, ykm2, ukm1 og ukm2 for å kalkulere outputen yk. Filen do\_filter\_calc\_server.py bruker denne servicen for å gjøre filtreringen, lp\_filter.py kaller altså denne servicen ved å sette opp ServiceProxy til servicen.

Filteret fungerer slik at vi lager en kontinuerlig transfer funksjon Hs som vi konverterer til en diskret transfer funksjon ved hjelp av funksjonen c2d i matlab. Dette gjøres ved å bruke Hs, sampling perioden 1/20 og zoh (Zero order hold) som input til funksjonen. Koeffisientene a1, a2, b1 og b2 er det vi får ut av vår nye funksjon Hdz. Disse brukes da for å filtrere signalet i funksjonen y[k] = -a1\*y[k-1]-a2\*y[k-2]+b1\*u[k-1]+b2\*u[k-2]

Cut-off frequency er definert der vi definerer omega\_0 i matlab koden der vi setter omega\_0 = 2pi\*0.4, her er tallet vi ganger 2pi med vår cut-off frequency. I dette tilfellet er det 0.4Hz. Sampling frequency er definert som 20Hz, og dermed satt til 1/20 i sampling perioden vår.