function [] = get_TF(Transfer_Functions)

Contents

```
function [H] = get_TF(Transfer_Functions)
verbose = 1;

syms s

signpost(verbose,'Start: get_TF()')
```

Initialise variables

```
signpost(verbose,'Initialise variables')

syms a1 da1 dda1
syms a2 da2 dda2
syms a3 da3 dda3

syms A1 A2 A3
syms tf1_a1_T1 tf2_a2_T2 tf3_a3_T3

syms l1 l2 l3

syms m1 m2 m3

syms Ixx1 Ixx2 Ixx3
syms Iyy1 Iyy2 Iyy3
syms Izz1 Izz2 Izz3

syms T1 T2 T3
```

Materialise

```
signpost(verbose,'Create Real Values')

Transfer_Functions = materialise(Transfer_Functions);
```

Collect s

```
signpost(verbose,'Collect s')

for i = 1:3
    tran(i,1) = rhs(collect(Transfer_Functions(i,1), s));
end
```

Find coefficients of polynomial

```
signpost(verbose,'Find coefficients of polynomial')

for i = 1:3
    cn = 0;
    cd = 0;
    if (tran(i,1) ~= 0)
        [n, d] = numden(tran(i,1));
        [cn, tn] = coeffs(n, s, 'all');
        [cd, td] = coeffs(d, s, 'all');
    end
    eq_n(i,:) = cn;
    eq_d(i,:) = cd;
end
```

Creating Transfer Functions from polynomial

```
signpost(verbose,'Creating Transfer Functions from polynomial')

for i = 1:3
    [n, d] = numden(tran(i,1));
    n = sym2poly(n);
    d = sym2poly(d);
    H(i,1) = tf(n,d, 'OutputName', strcat('ang_',int2str(i),' / torque_',
int2str(i)));
end
```

Tidy Up

```
signpost(verbose,'Done: get_TF()')
end
```

Published with MATLAB® R2017b