Supplementary Materials

The Matlab codes used for modeling are as:

```
clear all
close all
clc
%%
syms  Mh Jh           % Mass and Moment Inertia of Trunk
syms  Ms_int Js_int   % Mass and Moment of inertia of torque sensor (Internal Ring)
syms  Ms_ext Js_ext   % Mass and Moment of inertia of torque sensor (External Ring)
syms  M_act J_act     % Mass and Moment Inertia of Actuator
syms  g               % Gravity acceleration
syms  rh              % Raidal Displacement of Central mass of trunk with respect rotational axis
syms  Kh dh dhs       % Trunk Stiffness, Trunk Damping, Trk_trq Damping
syms  Ks ds           % Stiffness of torque sensor, Damping of torque sensor
syms  Kg dg dgs       % Gearbox Stiffness, Gearbox Damping, Gea_Trq Damping
syms  teta_h tetad_h tetadd_h          % Trunk Position, Trunk Velocity, Trunk Acceleration with respect to vertical axis
syms  teta_ext tetad_ext tetadd_ext    % torque sensor Position, Velocity, Acceleration (external ring)
syms  teta_int tetad_int tetadd_int    % torque sensor Position, Velocity, Acceleration (internal ring)
syms  teta_act tetad_act tetadd_act    % BDC motor shaft Position, BDC motor shaft Velocity, BDC motor shaft Acceleration, data from encoder
syms  tau_h tau_m                      % Torque by Trunk, Torque by BDC motor
syms  rg              % Gearbox ratio
syms  bm              % Friction between motor and gearbox
syms  Tm              % Motor Torque

%% Defining auxiliary variables
syms TETA_H TETAD_H TETADD_H
syms TETA_EXT TETAD_EXT TETADD_EXT
syms TETA_INT TETAD_INT TETADD_INT
syms TETA_ACT TETAD_ACT TETADD_ACT

%% Equations of Motion
% eqns = [ Tm   Bf*tetadm   Jtot*TETADDM   (Kg/rg)*( (tetam/rg)   tetas )   (Dg/rg)*( (tetadm/rg)   tetads ) == 0 ,
%          Kg*( (tetam/rg)   tetas ) + Dg*( (tetadm/rg)   tetads )   Js*TETADDS   Ks*(tetas   tetal)   Ds*(tetads   tetadl) == 0 ,
%          Ks*(  tetas   tetal ) + Ds*( tetads   tetadl )   Jl*TETADDL    Kl*(tetal   tetah)   Dl*(tetadl   tetadh) == 0  ];

eqns = [ Tm   bm*tetad_act   J_act*TETADD_ACT   (Kg/rg)*( (teta_act/rg)   teta_int )   (dgs/rg)*( (tetad_act/rg)   tetad_int )   (dg/rg)*(tetad_act/rg) == 0 ,
         Kg*( (teta_act/rg)   teta_int ) + dgs*( (tetad_act/rg)   tetad_int )   Js_int*TETADD_INT   Ks*(teta_int   teta_ext)   ds*(tetad_int   tetad_ext) == 0 ,
         Ks*(teta_int   teta_ext) + ds*(tetad_int   tetad_ext)   Js_ext*TETADD_EXT   Kh*(teta_ext   teta_h)   dhs*(tetad_ext   tetad_h) == 0 ,
         Kh*(teta_ext   teta_h) + dhs*(tetad_ext   tetad_h)   TETADD_H*Jh   tetad_h*dh == 0 ];

vars   = [ TETADD_ACT TETADD_INT TETADD_EXT TETADD_H ];
%%
eqns_mat = [ eqns ,
             TETAD_ACT == tetad_act ,
             TETAD_INT == tetad_int ,
             TETAD_EXT == tetad_ext ,
             TETAD_H   == tetad_h ];
vars_mat = [ TETAD_ACT TETADD_ACT TETAD_INT TETADD_INT TETAD_EXT TETADD_EXT TETAD_H TETADD_H];
[TETAD_ACT TETADD_ACT TETAD_INT TETADD_INT TETAD_EXT TETADD_EXT TETAD_H TETADD_H] = solve(eqns_mat, vars_mat)

%%
%STATE SPACE MODEL
%states = [ Motor position, Motor velocity, Internal torque sensor position, Internal torque sensor velocity, External torque sensor position, ...
%           External torque sensor velocity, Trunk position ]
%inputs = [ Motor torque, Trunk velocity ]

eqns_mat_form = [ TETAD_ACT TETADD_ACT TETAD_INT TETADD_INT TETAD_EXT TETADD_EXT TETAD_H ];
vars_mat_form = [ teta_act tetad_act teta_int tetad_int teta_ext tetad_ext teta_h];
ns = length(vars_mat_form);     % Number of States
[A,nB] = equationsToMatrix( eqns_mat_form , vars_mat_form );
A_eq = A;
Bt =  nB;
[B_eq,~] = equationsToMatrix( Bt , [ Tm , tetad_h  ] );
C_eq = [ 0 , 0  , Ks , ds ,  Ks ,  ds, 0  ]  ;
D_eq = [ 0 , 0 ];

%% Transfer Function
syms s
% H(s) = Y(s) / U(s) = C*(sI A)^(1)*B + D
disp('␣')
disp('The␣symbolic␣transfer␣function␣is:')
TF = simplify(  C_eq*[ (s*eye(ns,ns)   A_eq )^(1) ]*B_eq  + D_eq , 'Steps',100 );
pretty(TF)
disp('␣')
```