```matlab
function u = controller(z, param, x_des, dx_des, ddx_des)
   % ******** Implement your controller ********
   keypoints = keypoints_pend(z, param);
   rB = keypoints(:,2);
   err_position = x_des - rB;

   vB = velocity_rB(z, param);
   err_velocity = dx_des - vB;

   J_B = Jacobian_rB(z, param);

   Kp = 50;
   Kd = 5;
%    command = (ddx_des + Kp*err_position + Kd*err_velocity); % command for Lambda
   command = (Kp*err_position + Kd*err_velocity); % for last ctrl law

   % Oscillation
   dim = length(z);
   M = A_pend(z, param);
   z_vel_zero = z;
   z_vel_zero(dim/2+1:end) = zeros(dim/2, 1);
   u_zero = zeros(size(command));
   grav = -b_pend(z_vel_zero, u_zero, param);
   coriolis = -b_pend(z, u_zero, param) - grav;
   Jdot = Jdot_rB(z, param);

   Lambda_inv = J_B*inv(M)*J_B.';
   Lambda = inv(Lambda_inv);
   mu = Lambda*J_B*inv(M)*coriolis - Lambda*Jdot*z(dim/2+1:end);
   rho = Lambda*J_B*inv(M)*grav;

   %% Force commands
   F = Lambda*command + mu + rho;
%    F = Lambda*command + mu;
%    F = Lambda*command + rho;

   u = J_B.'*F;
end

```