Problem 1: The whole story of this one is in the script. All that is required is to build up the terminal impedance, find terminal current, do a divider to find rotor current $I_2$, then torque simply falls out as

$$T_m = 3\frac{p}{\omega} |I_2|^2 \frac{R_2}{s}$$

It is important to note that I am using RMS amplitude (and thus the multiplier is 3, not $\frac{3}{2}$). The other important thing to note is that we need to use phase voltage, NOT line-line voltage. A MATLAB script is appended.

Torque for this problem and for problem 3 are both plotted in Figure 1. The constant rotor resistance plot is the lower curve.

![Comparison of torques](image_url)

Figure 1: Torque-Speed Curves for Problems 1 and 3
Problem 2: This one, too, is mostly contained in the appended script. Construction of the ladder circuit (one ‘rung’ per slice) is described in the memo that was made available along with the problem set. To find slot impedance what we do is start at the right and note that the impedance to the left of the rightmost (that is lowest) inductance is:

\[ Z_1 = R_1 + j\omega L_1 \]

Then we put another resistance in parallel and an inductance in series:

\[ Z_2 = Z_1 \parallel (R_2 + j\omega L_2) \]

This process continues until we get to the top of the bar (the left-hand end of the ladder). And that is the answer. We have two plots to report: the first (Figure 2) is the bode plot. The second (Figure 3) is resistance vs. frequency. Note that there is a substantial increase of bar resistance from one Hz to 60 Hz, so we would expect higher starting performance, which we see in Figure 1.

![Graphs showing slot impedance and start performance](image)

Figure 2: Bode Plot of Slot Impedance

Problem 3 The solution to this one is actually in the same script as the solution to Problem 1. It could have been attacked in a number of ways. For example, you could have computed a vector of rotor frequencies corresponding to the vector of slip and simply used that in the
same calculation as you used for Problem 2. I converted that calculation into a function script and then used that to give me rotor resistance. The only dicey part of this is to refer to the low frequency rotor resistance.

A better solution to this would have included rotor reactance as well, but that is beyond our scope.
% Torque-Speed Curve for an Induction Motor
% Assumes the classical model
% This is a single-circuit model
% Required parameters are R1, X1, X2, R2, Xm, Vt, Ns
% Assumed is a three-phase motor
% This thing does a motoring, full speed range curve
% Copyright 1994 - 2003, James L. Kirtley Jr.
% Modified to do a comparison for 6.685 Problem Set 5
% ----------------------------------------------------------

X1 = 1.15;
X2 = 3.4;
Xm = 116;
R1 = .5;
R2b = .3;
p = 2;
f = 60;
Vt = 277;

% this stuff is for the slot geometry calculation
sig = 6e7;
w = .001 .* [ 1.0000 1.0694 1.1389 1.2083 1.2778 1.3472 1.4167 ...
       1.4861 1.5556 1.6250 1.6944 1.7639 1.8333 1.9028 ...
       1.9722 2.0417 2.1111 2.1806 2.2500 2.3194 2.3889 ...
       2.4583 2.5278 2.5972 2.6667 2.7361 2.8056 2.8750 ...
       2.9444 3.0139 3.0833 3.1528 3.2222 3.2917 3.3611 ...
       3.4306 3.5000 2.25 1.0 1.0 1.0 1.0 1.0 1.0 ...
       1.0000 1.75 2.5 3.25 4.0 3.25 2.5 1.75 1.0];
d = .0005;

Ns = 60*f/p;  % Synchronous Speed, RPM
oms = 2*pi*Ns/60;  % Synchronous speed, rad/sec
s = logspace(-3, 0, 100);  % get a nice vector of slips

N = Ns .* (1 - s);  % Speed, in RPM

Rr = R2b ./ s;  % Rotor resistance
Zr = j*X2 + Rr;  % Total rotor impedance
Za = j*Xm .* Zr ./ (j*Xm + Zr);  % Air-gap impedance
Zt = R1 + j*X1 +Za;  % Terminal impedance
Ia = Vt ./ Zt;  % Terminal Current
I2 = Ia .* j*Xm ./ (j*Xm + Zr);  % Rotor Current
Pag = 3 .* abs(I2).^2 .* Rr;  % Air-Gap Power
Pm = Pag .* (1 - s);  % Converted Power
% Developed Torque
Trqb = Pag ./ oms;

R2a = zeros(size(s)); % just a placeholder
Rs0 = zslotf(0, sig, d, w);

for i = 1:length(s)
    om = oms .* s; % this is per-unit frequency at the rotor
    [Rsr, Xr] = zslotf(om(i), sig, d, w);
    R2a(i) = (Rsr/Rs0) * R2b;
end

Rr = R2a ./ s; % Rotor resistance
Zr = j*X2 + Rr; % Total rotor impedance
Za = j*Xm .* Zr ./ (j*Xm + Zr); % Air-gap impedance
Zt = R1 + j*X1 +Za; % Terminal impedance
Ia = Vt ./ Zt; % Terminal Current
I2 = Ia .* j*Xm ./ (j*Xm + Zr); % Rotor Current
Pag = 3 .* abs(I2).^2 .* Rr; % Air-Gap Power
Pm = Pag .* (1 - s); % Converted Power
Trqa = Pag ./ oms; % Developed Torque

figure(1)
plot(N, Trqa, N, Trqb)
title('Comparison of torques');
ylabel('Torque, N-m');
xlabel('RPM');
% Script for Problem 2
% Slot impedance for a variable width slot
% Copyright 2003 James L. Kirtley Jr.
% all of this will be per unit length
% Start with description of slot width:
sig = 6e7;

% this is a vector of slot width
w = .001 .* [ 1.0000 1.0694 1.1389 1.2083 1.2778 1.4167 ... 
    1.4861 1.5556 1.6250 1.6944 1.7639 1.8333 1.9028 ... 
    1.9722 2.0417 2.1111 2.1806 2.2500 2.3194 2.3889 ... 
    2.4583 2.5278 2.5972 2.6667 2.7361 2.8056 2.8750 ... 
    2.9444 3.0139 3.0833 3.1528 3.2222 3.2917 3.3611 ... 
    3.4306 3.5000 2.25 1.0 1.0 1.0 1.0 1.0 ... 
    1.0000 1.75 2.5 3.25 4.0 3.25 2.5 1.75 1.0];

d = .0005; % slice height
muzero = pi*4e-7;
f = logspace(0,3,300);
om = 2*pi .* f;

% working from bottom to top, build up impedance
% bottom section:
r = (1/(sig*d))*(2/(w(1)+w(2))); % resistance of bottom part
x = (muzero*d*2/(w(1)+w(2))) .* om; % reactance of bottom part
Z0 = r + j .* x; % complex impedance

for i = 2:length(w), % now build toward the top
    r = (1/(sig*d))*(2/(w(i-1)+w(i))); % resistance
    x = (muzero*d*2/(w(i-1)+w(i))) .* om; % reactance of section
    Z1 = Z0 + (j/2).*x; % intermediate
    Z2 = Z1 .* r ./ (Z1 + r); % put resistance in parallel
    Z0 = Z2 + (j/2).*x; % new Z0 for next stage
end

% now we are done and can look at the results
figure(1)
clf
subplot 211
loglog(f, abs(Z0))
title('6.685 Problem Set 5, Problem 2: Complex Slot')
ylabel('Ohms/meter')

subplot 212
semilogx(f, (180/pi).*angle(Z0))
ylabel('Angle, Degrees')
xlabel('Frequency, Hz')

figure(2)
loglog(f, real(Z0))
title('6.685 Problem Set 5, Problem 2: Complex Slot')
ylabel('Resistance, Ohms/m')
xlabel('Frequency, Hz')

function [R, X] = zslotf(om, sig, d, w)
% Calculates slot impedance per unit length
% for a variable width slot
% Copyright 2003 James L. Kirtley Jr.
% om is frequency (radians/second)
% sig is conductivity (S/m)
% d is slot vertical increment (m)
% w is a vector of slot width at positions spaced d apart

muzero = pi*4e-7;

% working from bottom to top, build up impedance
% bottom section:
r= (1/(sig*d))*(2/(w(1)+w(2)));
% resistance of bottom part
x = (muzero*d*2/(w(1)+w(2))) * om;
% reactance of bottom part
Z0s = r + j * x;
% complex impedance

for i = 2:length(w),
% now build toward the top
r= (1/(sig*d))*(2/(w(i-1)+w(i)));
% resistance
x = (muzero*d*2/(w(i-1)+w(i))) * om;
% reactance of section
Z1 = Z0s + (j/2) * x;
% intermediate
Z2 = Z1 .* r / (Z1 + r);
% put resistance in parallel
Z0s = Z2 + (j/2) * x;
% new Z0 for next stage
end
R = real(Z0s);
X = imag(Z0s);