```
function [sol] = mmSolve_linSys(A,b,V,prm_isol)
%-------------------------------------------------------------------------
% [機能]
%   CGNR法で線形方程式を解く
% [入力]
%   A           ヤコビアン
%   b           残差ベクトル
%   v_tmp       測定データ
%   prm_isol    ソルバのパラメータの構造体
%       alp         受信電界残差に基づく正則化制御係数
%   dth         方程式を解く残差精度
%   dn          方程式を解く最大繰り返し回数
% [出力]
%   sol         線形方程式の解
%-------------------------------------------------------------------------

fprintf('    Inverse Solution : ¥n');

%% ローカル変数
M = size(A,1);              % 方程式の数
N = size(A,2);              % 変数の数
dth=prm_isol.tol;
dn=prm_isol.maxiter;
for iP = 1:M/2
    v_tmp(iP*2-1) = real(V(iP));
    v_tmp(iP*2)   = imag(V(iP));
end
err_PHY=sqrt(sum(b.*conj(b))./sum(v_tmp.*conj(v_tmp)));
g=prm_isol.alp*(1/N*trace(A'*A))*err_PHY.^2;
Q=A'*A+g*eye(N);
DD=diag(Q);
DD=DD.^-1;
DD=diag(DD);
Qt=DD*Q;
c=A'*b;
ct=DD*c;
sol=bicgstab(Qt,ct,dth,dn);

end
```