```matlab
function K = MTGP_covSum(cov, hyp, x, z, i)

% covSum - compose a covariance function as the sum of other covariance
% functions. This function doesn't actually compute very much on its own, it
% merely does some bookkeeping, and calls other covariance functions to do the
% actual work.
%
% Copyright (c) by Carl Edward Rasmussen & Hannes Nickisch 2010-09-10.
%
% modified by Robert Duerichen
% 04/02/2014
%
% See also COVFUNCTIONS.M.

if numel(cov)==0, error('We require at least one summand.'), end
for ii = 1:numel(cov)                          % iterate over covariance functions
  f = cov(ii); if iscell(f{:}), f = f{:}; end   % expand cell array if necessary
  j(ii) = cellstr(feval(f{:}));                       % collect number hypers
end

if nargin<3                                    % report number of parameters
  K = char(j(1)); for ii=2:length(cov), K = [K, '+', char(j(ii))]; end, return
end
if nargin<4, z = []; end     % make sure, z exists

[n,D] = size(x);
%% these lines have to be added to be able to use Lab_covCC_chol_nD function
if size(x,2) > 1
    nL = max(x(:,end));
end

v = [];                 % v vector indicates to which covariance parameters belong
for ii = 1:length(cov), v = [v repmat(ii, 1, eval(char(j(ii))))]; end

if nargin<5                                                      % covariances
  K = 0; if nargin==3, z = []; end                              % set default
  for ii = 1:length(cov)                       % iteration over summand functions
    f = cov(ii); if iscell(f{:}), f = f{:}; end % expand cell array if necessary
    K = K + feval(f{:}, hyp(v==ii), x, z);            % accumulate covariances
  end
else                                                            % derivatives
  if i<=length(v)
    vi = v(i);                                    % which covariance function
    j = sum(v(1:i)==vi);                     % which parameter in that covariance
    f  = cov(vi);
    if iscell(f{:}), f = f{:}; end        % dereference cell array if necessary
    K = feval(f{:}, hyp(v==vi), x, z, j);                 % compute derivative
  else
    error('Unknown hyperparameter')
  end
end
```