Complex NSOLTのためのSaivDrパッケージ変更点(案)

ver. 1.00

2016/4/24 長山 知司

* +saivdr/+dictionary/+nsoltx/AbstOvsdLpPuFb2dSystem.m
  1. function getMatrixE0\_()
     + 初期行列の計算方法を変更．現在はDCT行列より基底画像を生成しているようである．修正方法は要検討．
* +saivdr/+dictionary/+nsoltx/AbstOvsdLpPuFb2dTypeISystem.m
  1. getAnalysisFilterBank\_()
     + パラメータが増加したのでpmMtxSet\_の扱い方を変更する．Horizontal extension, Vertical extensionのforループ中で直交行列Wとベクトルthetaを取得するコードを追加．Order1BuildingBlockTypeI.stepの引数にW,thetaを追加．
* +saivdr/+dictionary/+nsoltx/AbstOvsdLpPuFb2dTypeIISystem.m
  1. getAnalysisFilterBank\_()
     + Horizontal extension, Vertical extension中のパラメータをW,UからW,U,hW,hU,thetaに変更する(W,Uは役割が変わることに留意)．変更したパラメータをOrder1BuildingBlockTypeII.stepの引数に追加．
* +saivdr/+dictionary/+nsoltx/AbstOvsdLpPuFb3dSystem.m
  + - 詳細未確認．おそらく初期行列の計算方法を変更する必要あり．
* +saivdr/+dictionary/+nsoltx/AbstOvsdLpPuFb3dTypeISystem
* +saivdr/+dictionary/+nsoltx/AbstOvsdLpPuFb3dTypeIISystem
  + - すぐに使用するわけではないので詳細未確認．おそらくgetAnalysisFilterBank\_()を修正する必要あり．
* +saivdr/+dictionary/+nsoltx/OvsdLpPuFb2dTypeIVm0System.m
* +saivdr/+dictionary/+nsoltx/OvsdLpPuFb2dTypeIVm1System.m
* +saivdr/+dictionary/+nsoltx/OvsdLpPuFb3dTypeIVm0System.m
* +saivdr/+dictionary/+nsoltx/OvsdLpPuFb3dTypeIVm1System.m
  1. updateParamterMatrixSet\_()
     + パラメータ行列が２種類に増加したため要修正．更にバタフライ行列のパラメータthetaも考慮する必要があるので，実装を大幅に変更する必要があるかもしれない．
* +saivdr/+dictionary/+nsoltx/OvsdLpPuFb2dTypeIIVm0System.m
* +saivdr/+dictionary/+nsoltx/OvsdLpPuFb2dTypeIIVm1System.m
* +saivdr/+dictionary/+nsoltx/OvsdLpPuFb3dTypeIIVm0System.m
* +saivdr/+dictionary/+nsoltx/OvsdLpPuFb3dTypeIIVm1System.m
  1. updateParameteMatrixSet\_()
     + TypeIと同様な問題により要修正．
* +saivdr/+dictionary/+nsoltx/AbstNsoltCoefManipulator2d.m
* +saivdr/+dictionary/+nsoltx/AbstNsoltCoefManipulator3d.m
  + - このクラスの使用法をまだ理解していない．おそらく修正が必要であるが，確認した限りでは使用例が見つからなかった．そのため，修正の優先順位は低く設定する．
* +saivdr/+dictionary/+nsoltx/ChannelGroup.m
  + - 変更不要．
* +saivdr/+dictionary/+nsoltx/NsoltAnalysis2dSystem.m
* +saivdr/+dictionary/+nsoltx/NsoltAnalysis3dSystem.m
* +saivdr/+dictionary/+nsoltx/NsoltSynthesis2dSystem.m
* +saivdr/+dictionary/+nsoltx/NsoltSynthesis3dSystem.m
* +saivdr/+dictionary/+nsoltx/NsoltFactory.m
  + - 現段階では変更は不要と判断．下位クラスの結合テストが終了してから再度確認する．
* +saivdr/+dictionary/+nsoltx/+mexsrcs/Order1BuildingBlockTypeI.m
  1. stepImpl()
     + 引数にmtxW, thetaを追加．R=blkdiag(mtxW,mtxU)に変更．変数thetaを用いて行列C,Sを生成．processQ\_()の引数にC,Sを渡す．
  2. processQ\_()
     + 引数に行列C,Sを追加．butterfly\_()を削除して行列積に置き換える．
  3. butterfly\_()
     + バタフライ行列がパラメータ依存になり，あえて使用する利点が見つからなくなったため削除．
* +saivdr/+dictionary/+nsoltx/+mexsrcs/Order2BuildingBlockTypeII.m
  1. nPs, nPa, Is, Ia, IsPsGreaterThanPa
     + ComplexNSOLTでは対称・非対称アトムの分類が無くなるので，これらの変数は全て削除．代わりにTypeIと同様にnHalfChannels, Iを定義する．
  2. setupImpl()
     + 引数ps, paをpに置き換える． nHalfChannels=p, nChannels=2p+1, I=eye(p), とする．その他nPs, nPa, Is, Ia, IsPsGreaterThanPaに関する行を全て削除．
  3. stepImpl()
     + 引数にmtxHW,mtxHU,theta1,theta2を追加，この際mtxW,mtxUの役割が変わる．条件分岐は不要なので削除．processQo\_(), processQe\_()は新規に作成する関数processQ\_()に統一する．processQ\_()の引数にtheta1あるいはtheta2を渡す． Rについて，１度目はR=blkdiag(mtxW,mtxU),２度めはR=blkdiag(mtxHW,obj.I)\*blkdiag(obj.I,mtxHU)に書き換える．
  4. processQo\_(), processQe()
     + 前述の理由により削除．
  5. processQ\_()
     + 具体的な実装方法は未定．
  6. butterfly\_()
     + TypeIの場合と同様な理由で削除．
* +saivdr/+dictionary/+nsoltx/+mexsrcs/GradEvalSteps2d.m
* +saivdr/+dictionary/+nsoltx/+mexsrcs/GradEvalSteps3d.m
  + - スパース表現に関してはCNSOLT本体の実装が完了してから行うので，現段階では変更不要．
* +saivdr/+dictionary/+utility/PolyPhaseMatrix1d
* +saivdr/+dictionary/+utility/PolyPhaseMatrix2d
* +saivdr/+dictionary/+utility/PolyPhaseMatrix3d
  + - 村松先生から複素数対応を行えとの指示．おそらく変更せずとも使用可能なので，CNSOLTの実装が一段落ついた次第にテストを行う．
* その他のファイル
  + - CNSOLTの本体に直接影響しないと判断したため．基本的に修正はしない．しかし，CNSOLTの特性測定(要素画像の確認，周波数応答，振幅応答等)に必要な場合は適宜変更を行う．