9月10日DETミーティング

神岡での測定(10/21-25)

- 10pTrms/rHzの信号を雑音レベルが10pTrms/rHzの装置で検出する。
  - 1%のスペクトル推定精度、1Hz以上、p値~0.05で検出するのに必要な測定時間は〜６時間。
  - 坑外に出る時に測定させて放置すればOKか？工事は何時から何時まで行われているのか？

神岡での測定(10/21-25)

- Bartington社のMag-03のスペックやいつレンタルできるかなどを問い合わせた。
  - 月曜日に電話をした。在庫を確認してもらっている。
  - アウトプットはD-subということだったのでデータロガーには接続可能？

ノイズバジェット

- RanaからeLIGOのコードを入手。eLIGOのコードは伝達関数を手で書いている

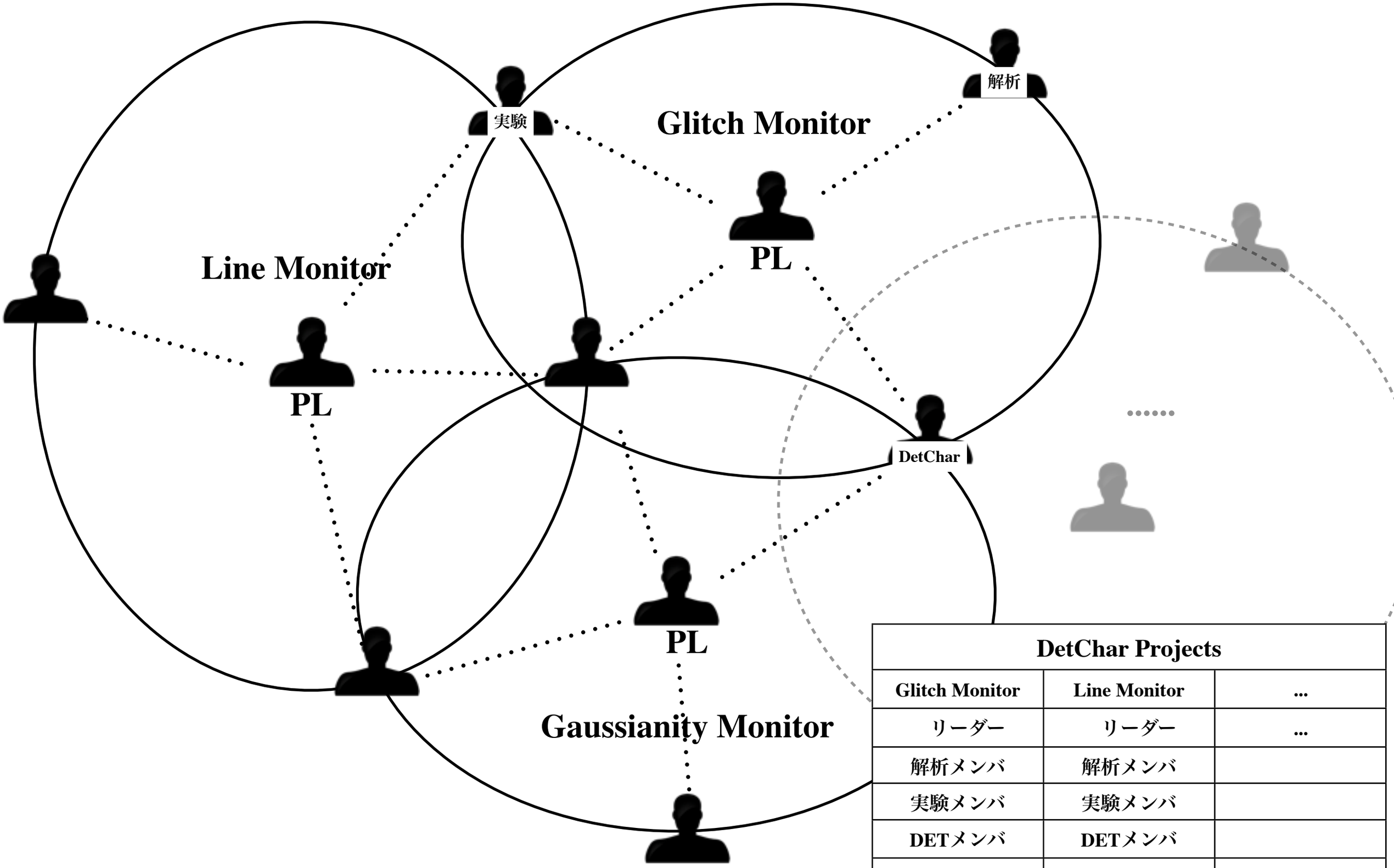
  https://dl.dropboxusercontent.com/u/66037561/eLIGO.zip
  に置いた。（二次配布はしないでください）

- aLIGOでは制御ループのモデリングにmatlab simlunkを用いて進めている。

  - 宮川、端山がメーリングリストに参加している。

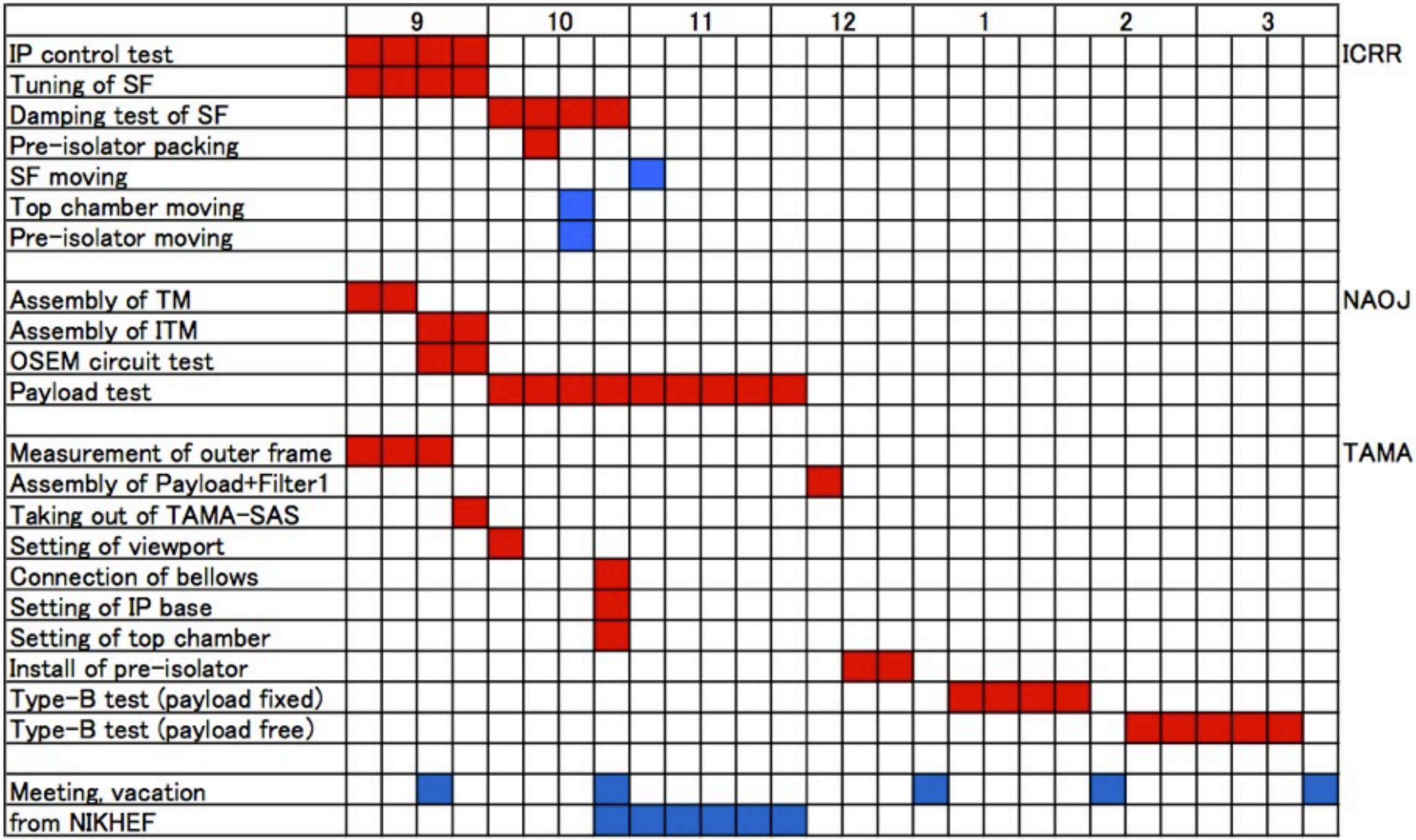
モニター等開発

プロジェクトの構成



DetChar Projects		
Glitch Monitor	Line Monitor	...
リーダー	リーダー	...
解析メンバ	解析メンバ	
実験メンバ	実験メンバ	
DETメンバ	DETメンバ	

VISの天文台でのテスト



- ラインモニタ
  - 伊藤さんに中心として開発
- グリッチモニターについて
  - まだ担当は決まっていない
  - まずはKleineWellパイプラインを実験の人が使える形にする
- Non-Gaussianityモニターについて
  - まだ担当は決まっていない
  - Rayleigh distributionモニタを実験の人が使える形にする
- ノイズバジェット
  - １、２ヶ月でコードを把握して、半年で開発。
  - CLIOでテストを行う。（スケジュールはまだ未定）

DetCharプロジェクト

Primary projectとSpecial projectに分け、P-が本筋で、S-は短期的または本筋に入れるために検討するプロジェクト。P-を優先して進め、S-は適時チームを結成して進める。
各プロジェクトにリーダーを配置し、中心になって進める。

Primary Projects

- Detcharシステムの構築(incl. web-based)
- Glitch Monitor
- Line Monitor
- Gaussianity Monitor
- Noise Budget
- Health Monitor
- Data base
- Quality flag

Special Projects

- Globally correlated noise
- Violin mode
- Multi-Channel Analysis
- Observation shift

チーム構成

各プロジェクトは

- リーダー
- 解析グループメンバ ≧ 1人
- 実験グループメンバ ≧ 1人
- DetCharメンバ ≧ 1人

で構成する

Glitch Monitor

- Glitch detection
- Statistics (frequency,..)
- Characterization
- Coherency check between channels
- Event display

Glitch Monitor開発に必要なもの

- Glitch detection
  - 基本的なアルゴリズムはExcess power method
  - Omega、Omicron、KleineWelle、 ExcessPower、GlitchMonなどがある。
- Statistics
  - Glitchの頻度、時間帯など
- Characterization
  - duration, central frequency, power
- Coherency check
  - ピアソンの積率相関関数など
- 解析結果の表示、グラフ

Line Monitor

- Line detection
- Statistics (frequency,..)
- Characterization (duration, central frequency, power)
- Coherency check between channels
- Event display

Line Monitor開発に必要なもの

- Line detection
  - 基本的なアルゴリズムはSFT、Monochromatic wave fitting、HHT(?)
  - Fscan、LineMon、MBLTなどがある。
- Statistics
  - Lineの頻度、時間帯など
- Characterization
  - duration, central frequency, power
- Coherency check
  - ピアソンの積率相関関数など
- 解析結果の表示、グラフ

Gaussianity Monitor

- Noise floor tracking
- Power spectrum
- Rayleigh distribution tracking
- Realtime noise modeling
- Monitor display

Gaussianity Monitor開発に必要なもの

- Noise floor tracking
  - Noise floorのスペクトルを追う
  - NoisefloorMonなどがある。
- Power spectrum
- Rayleigh distribution tracking
- Realtime noise modeling
  - まずはStudent-t分布によるフィッティング
- Monitor display

Noise Budget

- Tools for obtaining transfer functions
- Understanding noise budget tools in LIGO

DetChar system

- Platform to run the monitors automatically
  - DMT、Web-based displayなど
- Summary page generated automatically
  - 定期的にwebに公開する

Health Monitor

- To monitor important parameters of channels

チェックリストの活用

- ソフトウェアの開発はチェックリストを常時確認しながら行っていく。

チェックリスト（ツール開発）

- ツールの目的はなにか？
- 誰が使うのか？
- インプット、アウトプットフォーマットは規定に沿っているか？
- マニュアル、サンプルコードは用意しているか？
- コード内のコメントは修士１年生がわかるか？
- その箇所は共通ライブラリが使えないか？

チェックリスト（システムツール開発）

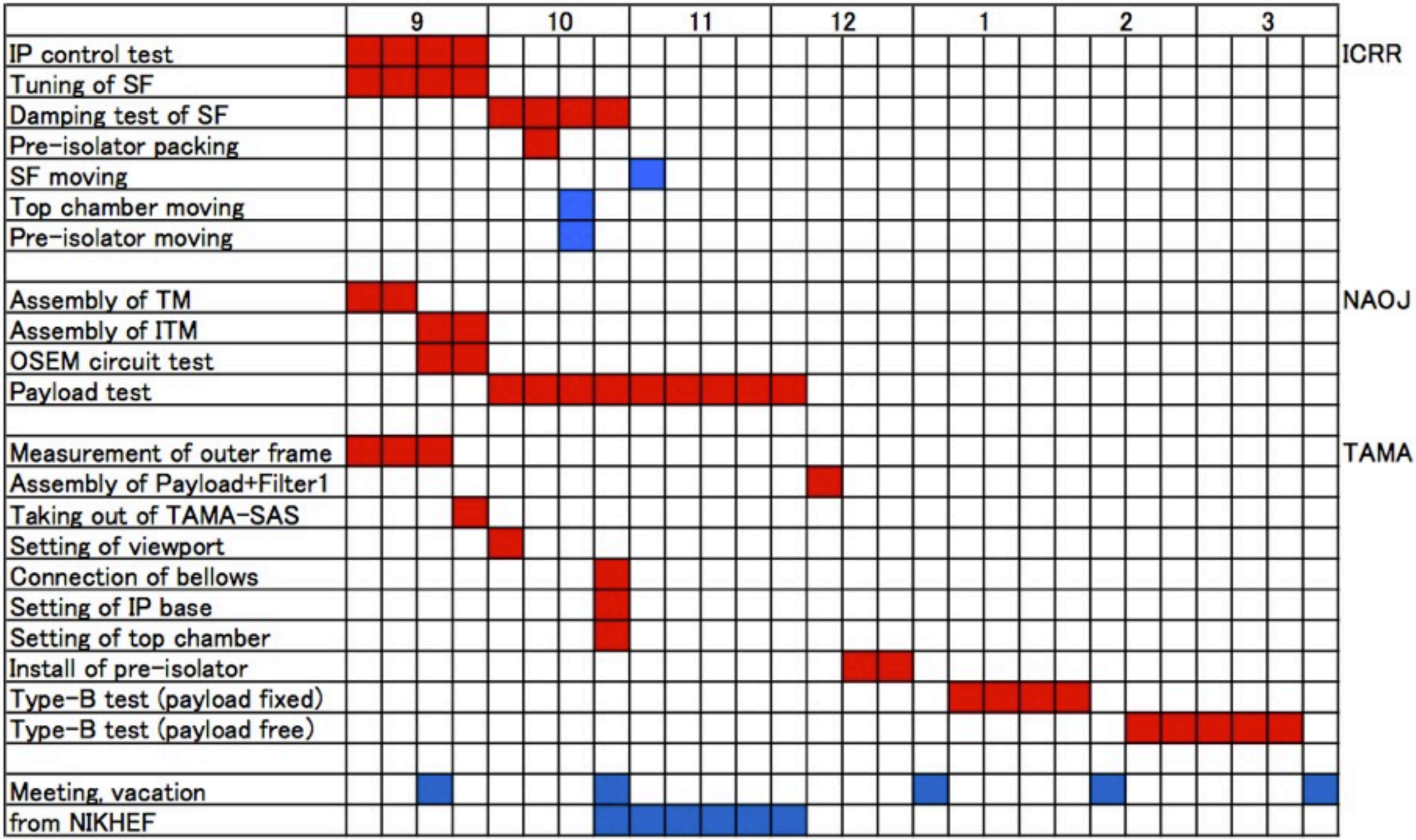
- ツールの目的は何か？
- 誰が使うのか？
- マニュアル、サンプルコードは用意しているか？
- リアルタイム解析できるだけ早いか？
- もっとシンプルにならないか？

タイムライン

近々の目標

- 秋の防振系テストでpreliminaryなツールが動かせること。
- 防振系テストで実験家に使ってもらいながらフィードバックをもらって改善する。

VISの天文台でのテスト



ルール、方針、心構え

- 常にチェックリストで確認しながら開発
- 人が作ったプログラムは使いにくいことを常に念頭に置いて、使う人の立場に立って開発
- プロジェクト間で共通で用いるアルゴリズム（プログラム）は、分かっているものについてはなるべく共有する。その場合、C/C++で書くのが良いだろう。

ツールのインターフェースについて

- インプットパラメータとアウトプットパラメータはどのツールも同じものにする。
- 途中の議論でインターフェースを変更する可能性もあることをあらかじめ承知しておく。