### Mapping Scheme

#### Design

Mapping Scheme klassens har til funktion af løse to opgaver

1. Lagre brugerindstillinger for hvordan data fra en given sensor omdannes til et MidiSignal
2. At syntetisere et MidiSignal ud fra en given sensor måling.

##### Klassediagram

Herunder ses et klassediagram for MappingScheme og dets lagringsstructs[[1]](#footnote-1) samt for MidiSignal, som MappingScheme også benytter i sin map() –funktion. Bemærk at der for klassediagrammet herunder er udeladt set- og get-metoder.



For funktionsbeskrivelser, se projektdokumentation s. XXX

#### Implementering

Herunder foreligger et udsnit af beskrivelsen af implementeringsfremgangen for klassens funktioner. For funktionerne mapVelocity, mapCCAbs og mapCCRel, se projektrapport s. XXX

Se bilag MappingScheme.h og MappingScheme.c for den endelige MappingScheme implementering.

##### map

map() har til ansvar at kalde den underfunktion, der svarer til den mapping parameter, brugeren har indstillet.

##### mapKey

Programflowet i mapKey er som vist i nedenstående flowchart



Tilrettelse af data: Systemet kan generere toner fra oktaverne -2 til 9 = 12 oktaver. I hver oktav er der 12 toner. Dette giver mulighed for at generer i alt 120 forskellige toner. Derfor benyttes kun sensorData i intervallet 3-122.

quantizeDiatonic: beskrives i flg. afsnit.

Dataforskydning muliggør korrekt kvantisering ved forskellige grundtoner.

Sæt Midisignal: For at muliggøre brug af flere note inputs på samme polyfoniske MIDI instrument, skal den forrige tone for sensoren slukkes inden en ny igangsættes. Der tjekkes derfor først om tonen er forskellig fra den gamle inden den sættes. Er den ny, sættes kommandoen til ”note off”. Dermed kræves to gennemløb med samme modtagne data før tonen ændres.

##### quantizeDiatonic

Herunder vises et udsnit af dokumentationen for quantizeDiatonic. Udover nedenstående, forefindes en beskrivelse af mol-kvantiseringen i projektrapporten s. XXX

Under et betegner man dur- og mol-skalaer som diatoniske skalaer – deraf navnet.

Når det indkomne data (som indeholder alle oktavens toner) skal kvantiseres, skiftes de toner, der ikke ligger i skalaen op- eller ned til en tone i skalaen. Kvantiseringen er designet således, at de mest brugte toner favoriseres: Grundtone (1. trin), terts (3. trin), kvint (5. trin), herunder markerede som hhv. 1, 3 og 5.

Dur kvantiseringen foregår som på illustrationen af en oktav herunder:



Påskrevet er durskalaens trin (1-7) for en c-dur skala samt hvilken vej det ønskes at tonerne uden for skala skal kvantiseres til (røde pile).

##### Test

Test af MappingScheme er for de fleste funktioner testede for et begrænset, men nøje udvalgt sæt input data. De er som udført af designer med udgangspunkt i en white box approach, hvor alle signalveje testes, om end ikke alle mulige udfald afprøves; dette ville for funktionen mapKey være *12 gruntoner x 3 skalaer x 2 retninger x 127 mulige datainput = 9144 muligheder* for testens udfald. Netop denne funktion er således testet for et begrænset antal inputs for en specifik grundtone-skala kombination. Der henvises til endelig integrationstest for auditiv validering af signalvej for forskellige grundtoner og skalaer.

Alle tests er at finde i projektdokumentationen s. XXX

1. Som specificeret i afsnittet ”Krav til Mapping Scheme” [↑](#footnote-ref-1)