Mise en œuvre du décodeur Matlab pour les flotteurs NKE Argos

Cette note concerne le décodeur Matlab des flotteurs Argos NKE PROVOR.

Les versions Coriolis concernées par cet outil sont 4.2, 4.21, 4.22, 4.23, 4.4, 4.41, 4.42, 4.43, 4.44, 4.45, 4.5 et 4.51.

[Mise en œuvre du décodeur Matlab pour les flotteurs NKE Argos 1](#_Toc384975913)

[1. Description de la livraison 2](#_Toc384975914)

[1.1. Répertoire « decPrv\_soft » 2](#_Toc384975915)

[1.1.1. Répertoire « config » 2](#_Toc384975916)

[1.1.2. Répertoire « soft » 2](#_Toc384975917)

[1.2. Répertoire « decPrv\_config\_floats » 2](#_Toc384975918)

[1.2.1. Répertoire « lists » 3](#_Toc384975919)

[1.2.2. Répertoire « prvFloatInfo » 3](#_Toc384975920)

[1.2.3. Répertoire « json\_float\_info » 3](#_Toc384975921)

[1.2.4. Répertoires « json\_float\_meta » et « json\_float\_meta\_remocean » 3](#_Toc384975922)

[1.2.5. Répertoire « DB\_export » 3](#_Toc384975923)

[2. Installation/configuration du décodeur 4](#_Toc384975924)

[3. Mise en œuvre du décodeur 5](#_Toc384975925)

[3.1. Principe de gestion des fichiers Argos 5](#_Toc384975926)

[3.1.1. Nomenclature des fichiers Argos cycle inexploités 7](#_Toc384975927)

[3.2. Programme decode\_provor\_2\_csv 8](#_Toc384975928)

[3.3. Programme decode\_provor\_2\_nc 9](#_Toc384975929)

[3.4. Programme decode\_provor\_2\_nc\_rt 10](#_Toc384975930)

[4. Décodage temps réel des flotteurs NKE Argos par le DAC Coriolis 12](#_Toc384975931)

[4.1. Déclaration des flotteurs 12](#_Toc384975932)

[4.2. Re-traitement de l’existant 12](#_Toc384975933)

[4.2.1. Mise en forme des données historiques 12](#_Toc384975934)

[4.2.2. Re-traitement des données historiques 13](#_Toc384975935)

[4.3. Traitement en routine 13](#_Toc384975936)

[4.4. Intégration d’un nouveau flotteur 13](#_Toc384975937)

[5. Description des paramètres de configuration 14](#_Toc384975938)

[6. Génération conditionnelle des fichiers NetCDF 17](#_Toc384975939)

# Description de la livraison

Le décodeur est livré dans un répertoire « decPrv\_*yyyymmdd* » qui contient :

* Un répertoire « decPrv\_soft » contenant le code matlab de l’application et les fichiers de configuration associés,
* Un répertoire « decPrv\_config\_floats » contenant les fichiers de configuration associés aux flotteurs pris en compte par le décodeur,
* Un répertoire « decPrv\_doc» contenant la documentation associée au décodeur.

## Répertoire « decPrv\_soft »

Il contient les répertoires « config » et « soft ».

### Répertoire « config »

Ce répertoire contient les fichiers de configuration associés à l’application :

* Le répertoire « configParamNames » : contenant le fichier de description des paramètres de configuration des flotteurs utilisés dans le fichier meta.nc (il y a un fichier par version de décodeur Matlab : **\_config\_param\_name \_*decoderID*.json**),
* Le répertoire « techParamNames » : contenant le fichier de description des paramètres techniques des flotteurs utilisés dans le fichier tech.nc (il y a un fichier par version de décodeur Matlab : **\_tech\_param\_name\_*decoderID*.json**),

### Répertoire « soft »

Il contient :

* Les 3 programmes permettant de mettre en œuvre le décodeur :
  + **decode\_provor\_2\_csv**,
  + **decode\_provor\_2\_nc**,
  + **decode\_provor\_2\_nc\_rt**,
* Les 2 fichiers de configuration principaux du décodeur :
  + « \_prv\_decoder.conf » (pour les programmes **decode\_provor\_2\_csv** et **decode\_provor\_2\_nc),**
  + « \_prv\_decoder\_conf.json » (pour le programme **decode\_provor\_2\_nc\_rt**),
* Le répertoire « sub » contenant les sous-programmes développés pour le décodeur,
* Le répertoire « sub\_foreign » contenant les sous-programmes utiles au décodeur et issus d’internet ou du LPO,
* Le répertoire « util » contenant quelques utilitaires externes au décodeur et le sous-répertoire « sub » contenant les sous-programmes associés.

## Répertoire « decPrv\_config\_floats »

Il contient les répertoires « lists », « prvFloatInfo », « json\_float\_info », « json\_float\_meta », « json\_float\_meta\_remocean » et « DB\_export ».

### Répertoire « lists »

Ce répertoire contient des listes de numéros WMO de flotteurs, elles sont utilisées via le paramètre de configuration FLOAT\_LIST\_FILE\_NAME des fichiers de configuration principaux du décodeur.

### Répertoire « prvFloatInfo »

Ce répertoire contient le fichier des paramètres de décodage des flotteurs déclarés : « prvFloatInfo/**\_provor\_floats\_information\_co.txt** »,

### Répertoire « json\_float\_info »

Ce répertoire contient les fichiers des paramètres de décodage des flotteurs susceptibles d’être sollicités par le décodeur (ex : **6901545\_122780\_info.json**).

Dans chaque fichier, si le paramètre DECODER\_ID est :

* Egal à -1 : Le flotteur est « ignoré », c'est-à-dire qu’il n’est pas pris en compte lors d’une tentative de décodage de ce flotteur (un message INFO associé est cependant généré),
* Différent de -1 : Le flotteur est pris en compte par le décodeur.

### Répertoires « json\_float\_meta » et « json\_float\_meta\_remocean »

Les répertoires « json\_float\_meta » et « json\_float\_meta\_remocean » contiennent les fichiers de méta-données des flotteurs déclarés. Il y a un fichier par flotteur (ex : **6901545\_meta.json**).

Rmq : Actuellement, le contenu de ces fichiers est généré à partir des données de la base Coriolis à l’exception du contenu des rubriques :

* CONFIG\_PARAMETER\_NAME,
* CONFIG\_PARAMETER\_VALUE,
* CONFIG\_MISSION\_NUMBER,
* SENSOR\_MOUNTED\_ON\_FLOAT,
* CALIBRATION\_COEFFICIENT,
* PRES\_SBE\_PUMP\_SWITCH\_OFF,
* SURF\_SLICE\_THICKNESS,
* RT\_OFFSET.

### Répertoire « DB\_export »

Ce répertoire contient, à titre d’information, l’export de la base Coriolis utilisé pour générer (via les outils generate\_json\_float\_info\_complement, generate\_json\_float\_meta et generate\_json\_float\_meta\_remocean) les fichiers de « json\_float\_info » des flotteurs « ignorés » par le décodeur, « json\_float\_meta » et « json\_float\_meta\_remocean » des flotteurs « déclarés ».

Le contenu de ce répertoire n’est donc pas directement utilisé par le décodeur.

# Installation/configuration du décodeur

Une fois le contenu du répertoire « decPrv\_*yyyymmdd* » copié sur la machine hôte :

* L’installation se fait en ajoutant au path matlab le répertoire « decPrv\_soft/soft » et ses sous-répertoires (les répertoires «  obsolete » peuvent être exclus de cette opération),
* La configuration du décodeur se fait en renseignant les paths des fichiers de configuration « decPrv\_soft/soft/**\_prv\_decoder.conf** » et « decPrv\_soft/soft /**\_prv\_decoder\_conf.json** » conformément aux répertoires d’installation sur la machine hôte.

Remarque : pour déployer le décodeur pour plusieurs utilisations (ou utilisateurs) seuls les fichiers de configuration nécessitent une duplication, puis une mise à jour selon le besoin.

Le fichier de configuration que l’on souhaite utiliser doit figurer en premier dans le path Matlab de chaque utilisation (ou utilisateur) (la commande Matlab ‘which \_prv\_decoder.conf’ ou ‘which \_prv\_decoder\_conf.json’ permet de s’assurer que le fichier de configuration utilisé est bien celui souhaité).

# Mise en œuvre du décodeur

Le décodeur peut être mis en œuvre de trois manières différentes, via les programmes :

* **decode\_provor\_2\_csv** : pour un traitement **temps différé** des données vers un fichier **CSV**,
* **decode\_provor\_2\_nc** : pour un traitement **temps différé** des données vers des fichiers **NetCDF**,
* **decode\_provor\_2\_nc\_rt** : pour un traitement **temps réel** des données vers des fichiers **NetCDF**.

Un traitement **temps différé** correspond à l’exploitation de tout ou partie de l’ensemble des fichiers Argos dont on dispose au moment du décodage.

Un traitement **temps réel** correspond à l’exploitation « au fil de l’eau » d’un flux de données.

Pour le décodeur Argos, ces deux modes de fonctionnement sont parfaitement compatibles (à condition de ne pas lancer simultanément la production d’un même fichier NetCDF). Ainsi peut-on, par exemple, effectuer un retraitement complet d’un flotteur avec **decode\_provor\_2\_nc** (en s’assurant toutefois au préalable qu’une instance de **decode\_provor\_2\_nc\_rt** n’est pas déjà en cours pour ce flotteur) puis de laisser la mise en œuvre automatique de **decode\_provor\_2\_nc\_rt** poursuivre le décodage des données qui arriveront par la suite.

## Principe de gestion des fichiers Argos

Les données Argos arrivent par mail et sont traitées par la collecte qui effectue le regroupement des données par identifiant de balise Argos.

La nomenclature des fichiers produits est :

*ArgosId*\_*yyyy-mm-dd-HH-MM-SS*.txt où *ArgosId* est l’identifiant Argos du flotteur (sur 6 digits) et *yyyy-mm-dd-HH-MM-SS* la date du premier (du plus ancien) message Argos flotteur contenu dans le fichier.

Chacun de ces fichiers doit contenir les données Argos relatives à un cycle et un seul ; aussi les appelle-t-on **fichiers Argos cycle**.

Le décodeur est alors mis en œuvre de manière automatique par la commande :

decode\_provor\_2\_nc\_rt('processmode', 'profile', 'argosfile', '*argosFilePath*/*ArgosId*\_*yyyy-mm-dd-HH-MM-SS*.txt’)

ou

decode\_provor\_2\_nc\_rt('processmode', 'all', 'argosfile', *argosFilePath*/*ArgosId*\_*yyyy-mm-dd-HH-MM-SS*.txt’)

où *argosFilePath* représente le path du fichier Argos à traiter

Le répertoire de stockage des fichiers Argos est défini par le paramètre de configuration DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1.

Dans ce répertoire, les données sont stockées par *ArgosId*; il y a ainsi un sous-répertoire, nommé *ArgosId* (sur 6 digits) qui regroupe toutes les données reçues avec cet identifiant Argos.

Au lancement d’une des commandes mentionnées ci-dessus, le décodeur effectue :

* La recherche du numéro WMO du flotteur concerné (*WMO*),
* La recherche du numéro du cycle concerné (*CycleNum*),
* La copie du fichier *argosFilePath*/*ArgosId\_yyyy-mm-dd-HH-MM-SS*.txt dans un répertoire temporaire DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId*/tmp,
* Le renommage du fichier *ArgosId\_yyyy-mm-dd-HH-MM-SS*.txt en *ArgosId\_yyyy-mm-dd-HH-MM-SS\_WMO\_CycleNum*.txt.
* La recherche, dans le répertoire DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId* d’un fichier de même cycle pour ce flotteur. Si un tel fichier existe, son contenu est concaténé au fichier DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId*/tmp/*ArgosId\_yyyy-mm-dd-HH-MM-SS\_WMO\_CycleNum*.txt.

Notons que ce cas correspond toujours à une anomalie, il peut par exemple être dû au fait :

* + D’un flotteur qui interrompt sa transmission Argos pendant un intervalle de temps inhabituel,
  + De CLS qui transmet avec un décalage anormal, les données collectées,
  + De la collecte dont le seuil temporel « de détection d’inactivité en transmission » du flotteur est trop court.
* Lorsque le paramètre 'processmode' vaut 'profile' :
  + Le traitement du fichier DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId*/tmp/*ArgosId\_yyyy-mm-dd-HH-MM-SS\_WMO\_CycleNum*.txt en vue de la génération du fichier NetCDF mono-profil associé.
* Lorsque le paramètre 'processmode' vaut 'all' :
  + Le déplacement du fichier DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId*/tmp/*ArgosId\_yyyy-mm-dd-HH-MM-SS\_WMO\_CycleNum*.txt vers le répertoire DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId*,
  + La suppression de l’éventuel fichier de même cycle pour ce flotteur dans le répertoire DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId*,
  + Le traitement du contenu du répertoire DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId* conformément au paramètre de configuration EXPECTED\_CYCLE\_LIST en vue de générer les fichiers NetCDF souhaités par les paramètres de configuration GENERATE\_NC\_TRAJ, GENERATE\_NC\_MULTI\_PROF, GENERATE\_NC\_MONO\_PROF, GENERATE\_NC\_TECH et GENERATE\_NC\_META.
  + Lorsque le traitement s’est déroulé correctement, la suppression du fichier *argosFilePath*/*ArgosId\_yyyy-mm-dd-HH-MM-SS*.txt.
* La suppression du répertoire DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId*/tmp et de son contenu éventuel.

**Un fichier Argos cycle n’est décodé que s’il est « identifié », c'est-à-dire s’il contient dans son nom les 3 informations : ArgosId, WMO et CycleNum.**

### Nomenclature des fichiers Argos cycle inexploités

Lorsque la recherche du numéro WMO échoue (cas d’un flotteur non déclaré) ou lorsque la détermination du numéro de cycle n’aboutit pas, le fichier n’est pas décodé.

Dans ce cas, et lorsque le paramètre 'processmode' vaut 'all', il est stocké dans le répertoire DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId* avec l’une des nomenclatures suivantes :

* *ArgosId\_yyyy-mm-dd-HH-MM-SS\_*WWWWWWW*\_*WWW.txt : numéro WMO indéterminé,
* *ArgosId\_yyyy-mm-dd-HH-MM-SS\_WMO\_*MMM.txt : numéro de cycle indéterminé par manque de méta-données,
* *ArgosId\_yyyy-mm-dd-HH-MM-SS\_WMO\_*TTT.txt : données de test émises avant la date de mise à l’eau du flotteur,
* *ArgosId\_yyyy-mm-dd-HH-MM-SS\_WMO\_*EEE.txt : données vides (pas un seul message Argos flotteur complet n’a pu être identifié),
* *ArgosId\_yyyy-mm-dd-HH-MM-SS\_WMO\_*GGG.txt : fichier *ghost*, constitué d’au plus 3 messages Argos flotteur.

Des outils (externes au décodeur) permettent à l’opérateur d’exclure volontairement certains fichiers Argos cycle du décodage.

Dans ce cas, ils sont stockés avec la nomenclature suivante :

*ArgosId\_yyyy-mm-dd-HH-MM-SS\_WMO\_*UUU.txt.

## Programme decode\_provor\_2\_csv

Les fichiers associés à ce programme sont :

* Entrées :
  + Fichier de configuration du décodeur : « **\_prv\_decoder.conf** »,
  + Fichier des paramètres de décodage des flotteurs déclarés : « decPrv\_config\_floats/prvFloatInfo/**\_provor\_floats\_information\_co.txt** »,
  + Fichiers de méta-données des flotteurs déclarés : ceux du répertoire « decPrv\_config\_floats/json\_float\_meta » (ex : **6901545\_meta.json**).
  + Fichiers Argos cycle d’entrée : ceux « identifiés » et stockés dans le répertoire DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId*.
* Sorties :
  + Fichier log créé dans le répertoire défini par le paramètre de configuration DIR\_OUTPUT\_LOG\_FILE,
  + Fichier CSV créé dans le répertoire défini par le paramètre de configuration DIR\_OUTPUT\_CSV\_FILE.

Paramètres d’entrée du programme :

* En l’absence d’argument, le décodage s’applique à tous les flotteurs de la liste définie par le paramètre de configuration FLOAT\_LIST\_FILE\_NAME,
* Il est cependant possible de préciser les flotteurs à décoder en fournissant en argument leur numéro WMO (ex : decode\_provor\_2\_csv(6901545) ou decode\_provor\_2\_csv(6901545, 6901546, 6901547)).

Les fichiers Argos cycle susceptibles d’être décodés sont ceux du répertoire DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId* qui ont été « identifiés », c'est-à-dire pour lesquels le numéro WMO du flotteur et le numéro du cycle apparaissent dans la nomenclature du fichier (i.e. l’attribution des données d’un cycle à un flotteur se fait uniquement sur le nom du fichier Argos cycle).

Parmi ces fichiers, seuls ceux répondant aux critères exprimés par le paramètre de configuration EXPECTED\_CYCLE\_LIST sont décodés.

## Programme decode\_provor\_2\_nc

Les fichiers associés à ce programme sont :

* Entrées :
  + Fichier de configuration du décodeur : « **\_prv\_decoder.conf** »,
  + Fichier des paramètres de décodage des flotteurs déclarés : « decPrv\_config\_floats/prvFloatInfo/**\_provor\_floats\_information\_co.txt** »,
  + Fichier de description des paramètres de configuration des flotteurs « decPrv\_soft/config/configParamNames/**\_config\_param\_name \_*decoderID*.json** »,
  + Fichier de description des paramètres techniques des flotteurs « decPrv\_soft/config/techParamNames/**\_tech\_param\_name\_*decoderID*.json** »,
  + Fichiers de méta-données des flotteurs déclarés : ceux du répertoire « decPrv\_config\_floats/json\_float\_meta » (ex : **6901545\_meta.json**),
  + Fichiers Argos cycle d’entrée : ceux « identifiés » et stockés dans le répertoire DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId*.
* Sorties :
  + Fichiers log créé dans le répertoire défini par le paramètre de configuration DIR\_OUTPUT\_LOG\_FILE,
  + Fichier NetCDF créés dans le répertoire défini par le paramètre de configuration DIR\_OUTPUT\_NETCDF\_FILE.

Les fichiers NetCDF générés sont ceux demandés par les flags :

* + - GENERATE\_NC\_TRAJ,
    - GENERATE\_NC\_MULTI\_PROF,
    - GENERATE\_NC\_MONO\_PROF,
    - GENERATE\_NC\_TECH,
    - GENERATE\_NC\_META.

du fichier « **\_prv\_decoder.conf** ».

Paramètres d’entrée du programme :

* En l’absence d’argument, le décodage s’applique à tous les flotteurs de la liste définie par le paramètre de configuration FLOAT\_LIST\_FILE\_NAME,
* Il est cependant possible de préciser les flotteurs à décoder en fournissant en argument leur numéro WMO (ex : decode\_provor\_2\_csv(6901545) ou decode\_provor\_2\_csv(6901545, 6901546, 6901547)).

Les fichiers Argos cycle susceptibles d’être décodés sont ceux du répertoire DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId* qui ont été « identifiés », c'est-à-dire pour lesquels le numéro WMO du flotteur et le numéro du cycle apparaissent dans la nomenclature du fichier (i.e. l’attribution des données d’un cycle à un flotteur se fait uniquement sur le nom du fichier Argos cycle).

Parmi ces fichiers, seuls ceux répondant aux critères exprimés par le paramètre de configuration EXPECTED\_CYCLE\_LIST sont décodés.

## Programme decode\_provor\_2\_nc\_rt

Les fichiers associés à ce programme sont :

* Entrées :
  + Fichier de configuration du décodeur : « **\_prv\_decoder\_conf.json**»,
  + Fichiers des paramètres de décodage des flotteurs « déclarés » ou « ignorés » : ceux du répertoire « decPrv\_config\_floats/json\_float\_info » (ex : **6901545\_122780\_info.json**),
  + Fichier de description des paramètres de configuration des flotteurs « decPrv\_soft/config/configParamNames/**\_config\_param\_name \_*decoderID*.json** »,
  + Fichier de description des paramètres techniques des flotteurs « decPrv\_soft/config/techParamNames/**\_tech\_param\_name\_*decoderID*.json** »,
  + Fichiers de méta-données des flotteurs déclarés : ceux du répertoire « decPrv\_config\_floats/json\_float\_meta » (ex : **6901545\_meta.json**),
  + Fichier(s) Argos cycles d’entrée :
    - Lorsque le paramètre 'processmode' vaut 'profile' : le fichier Argos cycle d’entrée est celui fourni via le paramètre 'argosfile'. S’il peut être « identifié », il est décodé.
    - Lorsque le paramètre 'processmode' vaut 'all' : les fichiers Argos cycles d’entrée sont ceux « identifiés » et déjà stockés dans le répertoire DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId* auxquels vient s’ajouter (s’il peut être « identifié ») le fichier Argos cycle fourni via le paramètre 'argosfile'.
* Sorties :
  + Fichiers log créé dans le répertoire défini par le paramètre de configuration DIR\_OUTPUT\_LOG\_FILE,
  + Fichier NetCDF créés dans le répertoire défini par le paramètre de configuration DIR\_OUTPUT\_NETCDF\_FILE.
    - Lorsque le paramètre 'processmode' vaut 'all' : les fichiers NetCDF sont générés conformément aux flags :
      * GENERATE\_NC\_TRAJ,
      * GENERATE\_NC\_MULTI\_PROF,
      * GENERATE\_NC\_MONO\_PROF,
      * GENERATE\_NC\_TECH,
      * GENERATE\_NC\_META.

du fichier « **\_prv\_decoder\_conf.json** »,

* + - Lorsque le paramètre 'processmode' vaut 'profile' : seul le fichier NetCDF mono-profil est généré conformément au flag GENERATE\_NC\_MONO\_PROF = 2.
  + Fichier XML de rapport d’exécution créé dans le répertoire défini par le paramètre de configuration DIR\_OUTPUT\_XML\_FILE.

Paramètres d’entrée du programme :

* Paramètres **obligatoires** :
  + Le paramètre 'argosfile' doit **nécessairement** être fourni au programme. Il désigne le fichier Argos sur lequel on souhaite déclencher le traitement. Ce fichier, fourni avec son path, doit avoir la nomenclature suivante :

*ArgosId\_yyyy-mm-dd-HH-MM-SS*.txt

où *ArgosId* est l’identifiant Argos du flotteur et *yyyy-mm-dd-HH-MM-SS* la date du premier (du plus ancien) message Argos flotteur contenu dans le fichier.

* + Le paramètre 'processmode' doit **nécessairement** être fourni au programme. Les deux valeurs possibles de ce paramètre s sont 'all' et 'profile'.
    - Lorsque 'processmode' vaut 'all' :
      * Le traitement s’applique au fichier fourni via le paramètre 'argosfile' et à tous les fichiers Argos cycle « identifiés » du répertoire DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId* dont les numéros de cycles répondent aux critères exprimés par le paramètre de configuration EXPECTED\_CYCLE\_LIST,
      * A l’issue du traitement, le fichier fourni via le paramètre 'argosfile' est supprimé, il est stocké dans le répertoire DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId* avec la nomenclature suivante :

*ArgosId\_yyyy-mm-dd-HH-MM-SS\_WMO\_CycleNum*.txt

où *ArgosId* est l’identifiant Argos du flotteur, *yyyy-mm-dd-HH-MM-SS* la date du premier (du plus ancien) message Argos flotteur contenu dans le fichier, *WMO* est le numéro WMO du flotteur et *CycleNum* le numéro du cycle.

* + - Lorsque 'processmode' vaut 'profile' :
      * Le traitement s’applique au fichier fourni via le paramètre 'argosfile' lorsque son numéro de cycle, une fois déterminé, répond aux critères exprimés par le paramètre de configuration EXPECTED\_CYCLE\_LIST,
      * A l’issue du traitement, le fichier fourni via le paramètre 'argosfile' est inchangé, aucun stockage supplémentaire n’est effectué dans le répertoire DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId*.
* Paramètres **optionnels** :
  + Le paramètre 'configfile' peut être fourni au programme. Il désigne le fichier de configuration (avec son path complet) à utiliser en lieu et place du premier fichier « **\_prv\_decoder\_conf.json**» présent dans le path matlab,
  + Le paramètre 'xmlreport' peut être fourni au programme. Il désigne le nom du fichier XML de rapport d’exécution créé dans le répertoire défini par le paramètre de configuration DIR\_OUTPUT\_XML\_FILE.

La nomenclature de ce fichier doit être co041404\_*yyyymmddTHHMMSSZ*.xml où *yyyymmddTHHMMSSZ* désigne une date (qui sera également reprise dans le nom du fichier log créé dans le répertoire défini par le paramètre de configuration DIR\_OUTPUT\_LOG\_FILE),

* + Chaque paramètre du fichier « **\_prv\_decoder\_conf.json**» peut être remplacé par un paramètre de même nom passé en argument du programme.

Par exemple la commande

decode\_provor\_2\_nc\_rt('processmode', 'profile', 'argosfile', '*argosFilePath*/*ArgosId\_yyyy-mm-dd-HH-MM-SS*.txt’, ‘FLOAT\_TRANSMISSION\_TYPE’, ‘1’)

permet, en plus, de s’assurer que c’est bien le décodeur de données Argos qui est mis en œuvre (et ce quel que soit le contenu du fichier de configuration).

# Décodage temps réel des flotteurs NKE Argos par le DAC Coriolis

## Déclaration des flotteurs

Le flotteur 6901545 (ArgosId : 122780) est **déclaré** lorsque l’on a généré ses fichiers :

* « json/json\_float\_info/6901545\_122780\_info.json »,
* « json/json\_float\_meta/6901545\_meta.json».

## Re-traitement de l’existant

### Mise en forme des données historiques

Les données Argos reçues et traitées au centre Coriolis sont stockées dans le répertoire /home/coriolis\_exp/spool/co01/co0101/archive/cycle qui contient plus de 170 000 fichiers cycles.

Ces fichiers cycles « historiques » comportent quelques anomalies que nous proposons de corriger :

1. Ils sont tous stockés dans un même répertoire, ce qui rend leur manipulation peu pratique et la détection visuelle des anomalies de nomenclature difficile,
2. Il y a des anomalies dans les numéros de cycles,
3. Certains cycles comportent plus d’un fichier cycle,
4. Certains fichiers ne sont pas « identifiés » (détermination du numéro WMO du flotteur et du numéro du cycle) et donc inexploités alors qu’ils devraient l’être,
5. La nomenclature de certains fichiers a été volontairement modifiée pour en faire la sauvegarde et les extraire du traitement (ex : 66203\_2009-02-25\_7900117\_087\_25450\_KO.txt ou 112546\_2012-07-06\_6901022\_000X\_14032.txt),
6. Certains fichiers sont vides de données (ils ne contiennent même pas un message Argos flotteur) car créés à partir de *ghost messages* (ArgosId aléatoirement créés par le bruit électronique ambiant et identifiés par les satellites du système Argos).

Pour mettre en forme ces données « historiques » nous proposons tout d’abord d’effectuer (via l’utilitaire copy\_argos\_files) une copie des données Argos associées aux flotteurs NKE Argos concernés par le décodeur Matlab (473 flotteurs actuellement).

Puis d’en corriger (via l’utilitaire move\_and\_rename\_argos\_files) les anomalies, cette opération consistant à :

1. Stocker dans un même répertoire toutes les données reçues avec le même identifiant Argos,
2. Identifier plus précisément les numéros de cycles (en fonction des caractéristiques et de la configuration de chaque flotteur),
3. Regrouper toutes les données Argos d’un cycle donné dans un seul et unique fichier cycle,
4. Relancer l’identification de tous les fichiers,
5. Stocker dans un répertoire spécifique (pour les sauvegarder ?) les fichiers ne répondant pas à la nomenclature attendue,
6. Stocker dans un répertoire spécifique (pour les supprimer ?) les fichiers vides de données.

### Re-traitement des données historiques

Une fois mises en forme, les données historiques seront re-traitées avec le programme **decode\_provor\_2\_nc**.

## Traitement en routine

Une fois le re-traitement des données historiques effectué, le traitement temps réel des données reçues par e-mail peut être mis en œuvre par la commande

decode\_provor\_2\_nc\_rt('processmode', 'profile', 'argosfile', '*argosFilePath*/*ArgosId\_yyyy-mm-dd-HH-MM-SS*.txt’)

lorsque le flotteur est en cours de transmission Argos

ou

decode\_provor\_2\_nc\_rt('processmode', 'all', 'argosfile', *argosFilePath*/*ArgosId\_yyyy-mm-dd-HH-MM-SS*.txt’)

lorsque le flotteur a terminé la transmission Argos d’un cycle donné.

## Intégration d’un nouveau flotteur

L’intégration d’un nouveau flotteur se fait uniquement en le déclarant (cf. §1.5.1).

Dans l’hypothèse où le nouveau flotteur a déjà transmis des données, elles ont été stockées dans le répertoire DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1/*ArgosId* avec la nomenclature *ArgosId\_yyyy-mm-dd-HH-MM-SS\_*WWWWWWW*\_*WWW.txt.

Dans ce cas, il est au préalable nécessaire de mettre en forme ces données (i.e. de les identifier) avec l’utilitaire move\_and\_rename\_argos\_files puis de les traiter avec le programme **decode\_provor\_2\_nc**.

# Description des paramètres de configuration

Les paramètres de configuration du décodeur Matlab sont stockés dans le fichier de configuration « **\_prv\_decoder.conf** » pour les décodeurs temps différé (**decode\_provor\_2\_csv** et **decode\_provor\_2\_nc**) et dans le fichier de configuration « **\_prv\_decoder.json** » pour le décodeur temps réel (**decode\_provor\_2\_nc\_rt**).

Les paramètres que l’on peut trouver dans l’un ou l’autre de ces fichiers de configuration sont les suivants :

**FLOAT\_LIST\_FILE\_NAME :**

Désigne un fichier contenant une liste de numéros WMO de flotteurs.

Cette liste est utilisée comme liste de traitement par défaut lorsque l’on utilise les programmes **decode\_provor\_2\_csv** ou **decode\_provor\_2\_nc** sans paramètre ou le programme **decode\_provor\_2\_nc\_rt** pour traiter des flotteurs Remocean.

*Préconisation : doit contenir la liste des flotteurs déclarés.*

**EXPECTED\_CYCLE\_LIST :**

Fournit une expression permettant de déterminer les cycles que l’on souhaite décoder.

Les règles de cette expression sont les suivantes :

LIST OF CYCLES TO DECODE

given by:

an integer N => the first N existing cycles

[N1, N2, N3] => the given cycle numbers

[~] => all cycles

[N1~N2] => from cycle #N1 to cycle #N2

[~N2] => all cycles until cycle #N2

[N1~] => all cycles from cycle #N1

*Préconisation : mettre [~] ou 9999 de manière décoder tous les cycles disponibles.*

**FLOAT\_TRANSMISSION\_TYPE :**

Fournit le type de données à décoder (Argos, Iridium RUDICS ou Iridium SBD). Rappelons en effet que le décodeur ne peut traiter, lors d’une même instance, que des données de même type.

Le codage de cette information est le suivant :

FLOAT\_TRANSMISSION\_TYPE

1 for Argos transmission

2 for Iridium Rudics transmission

3 for Iridium SBD transmission

**FLOAT\_INFORMATION\_FILE\_NAME :**

Désigne un fichier contenant les paramètres de décodage d’un ensemble de flotteurs.

Ce fichier est utilisé par les programmes **decode\_provor\_2\_csv** ou **decode\_provor\_2\_nc**.

**DIR\_INPUT\_JSON\_FLOAT\_DECODING\_PARAMETERS\_FILE :**

Désigne le répertoire contenant les fichiers json des paramètres de décodage des flotteurs. Il y a un fichier json par flotteur (ex : 6901545\_122780\_info.json).

Ces fichiers json sont utilisés par le programme **decode\_provor\_2\_nc\_rt**.

**HEX\_ARGOS\_FILE\_FORMAT :**

Fournit le numéro du format des fichiers Argos à décoder.

Le codage de cette information est le suivant :

INPUT ARGOS RAW DATA FORMAT

format #1 => Argos PRV/DS format (original format from CLS)

format #2 => ARGOS-ARGO DEP format (format used for ANDRO atlas)

*Préconisation : mettre 1.*

**DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1 et DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_2 :**

Désigne le répertoire de stockage des données Argos à décoder au format #1 et #2 respectivement.

**HEX\_ARGOS\_DATA\_DIRECTORY\_STRUCTURE :**

Fournit la structure de stockage des données Argos à décoder.

Le codage de cette information est le suivant :

INPUT ARGOS RAW DATA DIRECTORY STRUCTURE

How Argos raw data files are stored in the DIR\_INPUT\_HEX\_ARGOS\_FILE\_FORMAT\_1(2) directory?

structure #1 => one directory (named "argosId\_wmo") per float

structure #2 => all argos raw data in the same directory

structure #3 => one directory (named "argosId") per Argos Id

*Préconisation : mettre 3 (ou 2 si l’on souhaite utiliser les données « historiques » archivées dans le répertoire /home/coriolis\_exp/spool/co01/co0101/archive/cycle).*

**DIR\_INPUT\_RSYNC\_DATA :**

Désigne le répertoire contenant les données Iridium RUDICS collectées via rsync.

Cette information est exploitée par le programme **decode\_provor\_2\_nc\_rt** pour traiter des flotteurs Remocean.

**DIR\_INPUT\_RSYNC\_LOG :**

Désigne le répertoire contenant les rapports de collecte des données Iridium RUDICS collectées via rsync.

Cette information est exploitée par le programme **decode\_provor\_2\_nc\_rt** pour traiter des flotteurs Remocean.

**DIR\_INPUT\_JSON\_TECH\_LABEL\_FILE :**

Désigne le répertoire contenant les fichiers json de description des paramètres techniques des flotteurs. Il y a un fichier json par version de décodeur matlab (\_tech\_param\_name\_*decoderID*.json)

**DIR\_INPUT\_JSON\_CONF\_LABEL\_FILE :**

Désigne le répertoire contenant les fichiers json de description des paramètres de configuration des flotteurs. Il y a un fichier json par version de décodeur matlab (\_config\_param\_name\_*decoderID*.json)

**DIR\_INPUT\_JSON\_FLOAT\_META\_DATA\_FILE :**

Désigne le répertoire contenant les fichiers json des méta-données des flotteurs. Il y a un fichier json par flotteur (ex : 6901545\_meta.json).

**IRIDIUM\_DATA\_DIRECTORY :**

Désigne le répertoire de stockage des données Iridium RUDICS à décoder.

**DIR\_OUTPUT\_LOG\_FILE :**

Désigne le répertoire de stockage des fichiers log générés par le décodeur. Un fichier log est généré lors de chaque instance du décodeur.

**DIR\_OUTPUT\_CSV\_FILE :**

Désigne le répertoire de stockage des fichiers CSV générés par le programme **decode\_provor\_2\_csv**. Un fichier CSV est généré lors de chaque instance de ce programme.

**DIR\_OUTPUT\_XML\_FILE :**

Désigne le répertoire de stockage des rapports de traitement XML générés par le programme **decode\_provor\_2\_nc\_rt**. Un fichier XML est généré lors de chaque instance de ce programme.

**DIR\_OUTPUT\_NETCDF\_FILE :**

Désigne le répertoire de stockage des fichiers NetCDF générés par les programmes **decode\_provor\_2\_nc** et **decode\_provor\_2\_nc\_rt**.

**GENERATE\_NC\_TRAJ, GENERATE\_NC\_MULTI\_PROF, GENERATE\_NC\_MONO\_PROF, GENERATE\_NC\_TECH et GENERATE\_NC\_META :**

Flags permettant de préciser les fichiers NetCDF générés par les programmes **decode\_provor\_2\_nc** et **decode\_provor\_2\_nc\_rt**.

Un flag à 0 signifie que l’on ne souhaite pas produire le fichier en question.

Un flag à 1 signifie que l’on souhaite toujours produire le fichier en question.

Un flag à 2 signifie que l’on souhaite produire le fichier en question sous condition (cf. § 1.7).

Le programme **decode\_provor\_2\_nc\_rt**, lorsqu’il est utilisé pour décoder des flotteurs Argos et que le paramètre 'processmode' vaut 'profile', positionne, pour la session courante, ces flags à 0 à l’exception de GENERATE\_NC\_MONO\_PROF qui est à mis à 2 et de GENERATE\_NC\_META qui est inchangé.

*Préconisation : mettre ces 5 flags à 2 pour un décodage temps réel.*

**ADD\_THREE\_MINUTES :**

Flag permettant de préciser si l’on souhaite ajouter 3 minutes à certaines informations temporelles fournies par le flotteur.

*Préconisation : mettre 0.*

**CYCLE\_2\_SBD\_DATA\_FILE\_NAME :**

Désigne le fichier d’association des numéros de cycles aux fichiers SBD des flotteurs Iridium SBD.

*Préconisation : peu importe (utilisée uniquement pour retraiter les données Iridium SBD en temps différé dans le cadre d’ANDRO).*

**FLOAT\_WITH\_DAMPING\_DISK\_FILE\_NAME :**

Désigne la liste des flotteurs équipés d’une collerette et pour lesquels il faut appliquer une correction spécifique aux données de conductivité mesurées par le flotteur.

*Préconisation : peu importe (il n’y a plus de flotteurs FSI actifs).*

# Génération conditionnelle des fichiers NetCDF

Les fichiers NetCDF de données de trajectoire (TRAJ), de données de profil (MONO-PROF ou MULTI-PROF), de données techniques (TECH) et de méta-données (META) sont générés conformément au contenu des flags de configuration GENERATE\_NC\_TRAJ, GENERATE\_NC\_MULTI\_PROF, GENERATE\_NC\_MONO\_PROF, GENERATE\_NC\_TECH et GENERATE\_NC\_META.

Les règles de génération de ces fichiers sont les suivantes :

* Lorsque le flag de génération d'un fichier NetCDF est à 0, il n'est pas généré,
* Lorsque le flag de génération d'un fichier NetCDF est à 1, il est systématiquement généré (créé ou mis à jour),
* Lorsque le flag de génération d'un fichier NetCDF est à 2 :
  + Le fichier méta-données est généré que s’il n’existe pas déjà (i.e. il n’est jamais mis à jour),
  + En temps réel et en mode 'profile' :
* Le flag de génération du fichier mono-profil est forcé à 2,
* Les flags de génération des fichiers multi-profils, trajectoire et technique sont forcés à 0.

Les règles de génération des fichiers NetCDF sont alors :

* Pour le fichier mono-profil : le fichier n'est créé que lorsque le profil est complet; une fois créé il n'est plus mis à jour,
* Pour les fichiers multi-profils, trajectoire et technique : pas de création ni de mise à jour,
  + En temps réel et en mode 'all' ou en temps différé :
    - Fichier mono-profil :
      * Le fichier est créé s'il n'existe pas déjà (i.e. si le profil est incomplet); dans ce cas, un message WARNING précise que le profil est incomplet,
      * Le fichier est mis à jour soit :
        + Lorsque la date système du fichier Argos cycle concerné est identique ou postérieure à la date de mise à jour du fichier mono-profil (nouvelles données Argos arrivées depuis la création du fichier mono-profil),
        + Lorsque le profil du fichier mono-profil n'est pas localisé (les données Argos des cycles suivants sont susceptibles de fournir après coup, par interpolation, une position à ce profil).
    - Fichiers multi-profils, trajectoire et technique :
      * Le fichier est créé s'il n'existe pas déjà,
      * Le fichier est mis à jour, sauf lorsque les deux conditions suivantes sont vérifiées :
        + Les cycles associés aux fichiers Argos cycle sont inclus dans le tableau des cycles du fichier NetCDF,
        + Aucune date système des fichiers Argos cycle n'est identique ou postérieure à la date de mise à jour du fichier NetCDF.