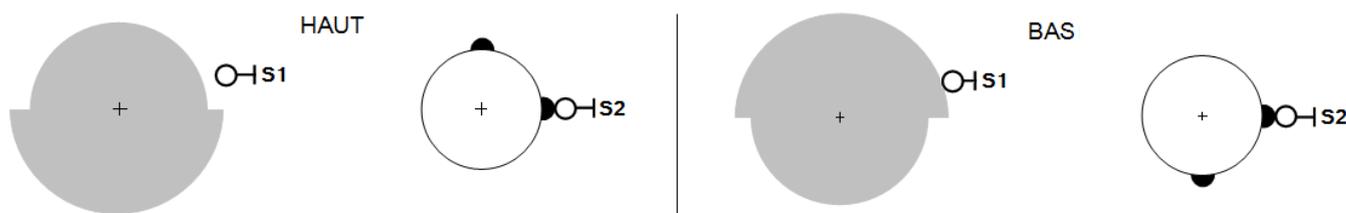


**VIGIPARK**



**1 / Informations délivrées par les détecteurs S1, S2**

**Q1 / COMPLÉTER** le tableau : **INDIQUER** le niveau logique des détecteurs S1, S2 selon la situation.



*Etat des détecteurs S1, S2 en fin de course*

	S1	S2
Haut		
Partie haute		
Partie Basse		
Bas		

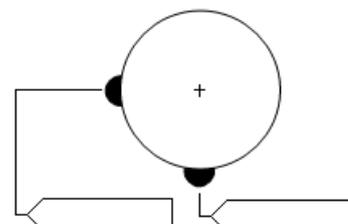
*Etats des détecteurs*

**Q2 / COMPLÉTER** le tableau : **PRÉCISER** le sens de rotation (CW/CCW) de l'arbre principal et de l'arbre réducteur selon le mouvement.

	Arbre principal	Arbre réducteur
Montée		
Descente		

*Sens de rotation des arbres*

**Q3 / PRÉCISER** la fonction respectivement assurée par les détecteurs S1 et S2. Sur la figure, **NOMMER** « plot\_h » et « plot\_b » les plots qui actionnent S2.



Les équations des variables « Haut » et « Bas » en fonction des détecteurs S1 et S2 s'écrivent donc :

$$\text{Haut} = \overline{S1} \cdot S2$$

$$\text{Bas} = S1 \cdot S2$$

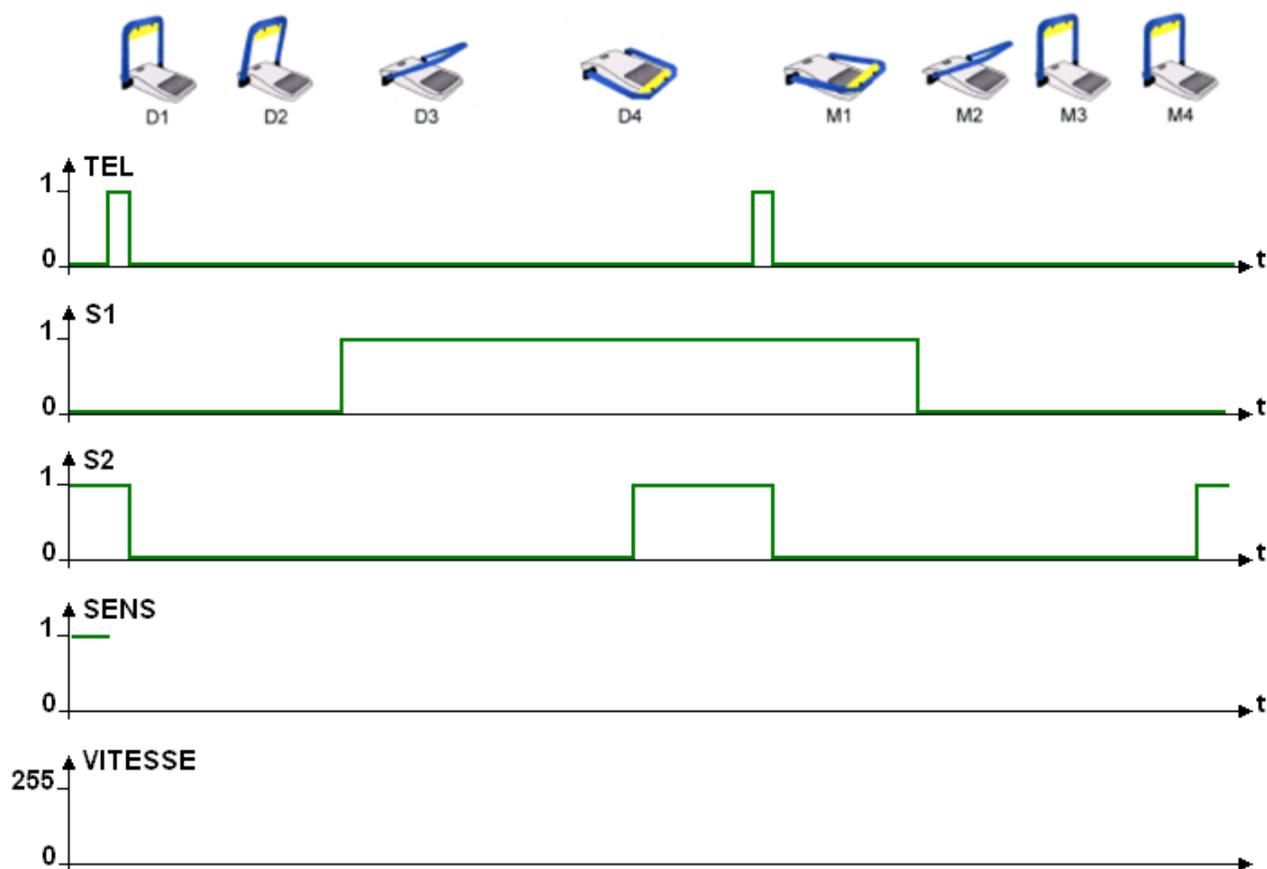
Le tableau suivant (issu du Dossier Technique) précise, pour chaque situation de l'arceau :

- le niveau logique à donner à la variable logique de sortie « SENS » (« 0 », « 1 », « X » = indifférent)
- la/les valeur(s) possible(s) à donner à la variable analogique de sortie « VIT ».

Situation de l'arceau	SENS	VIT
A l'arrêt	X	0
En montée	1	]0 ... 255]
En descente	0	]0 ... 255]

### 2 / Description temporelle du fonctionnement

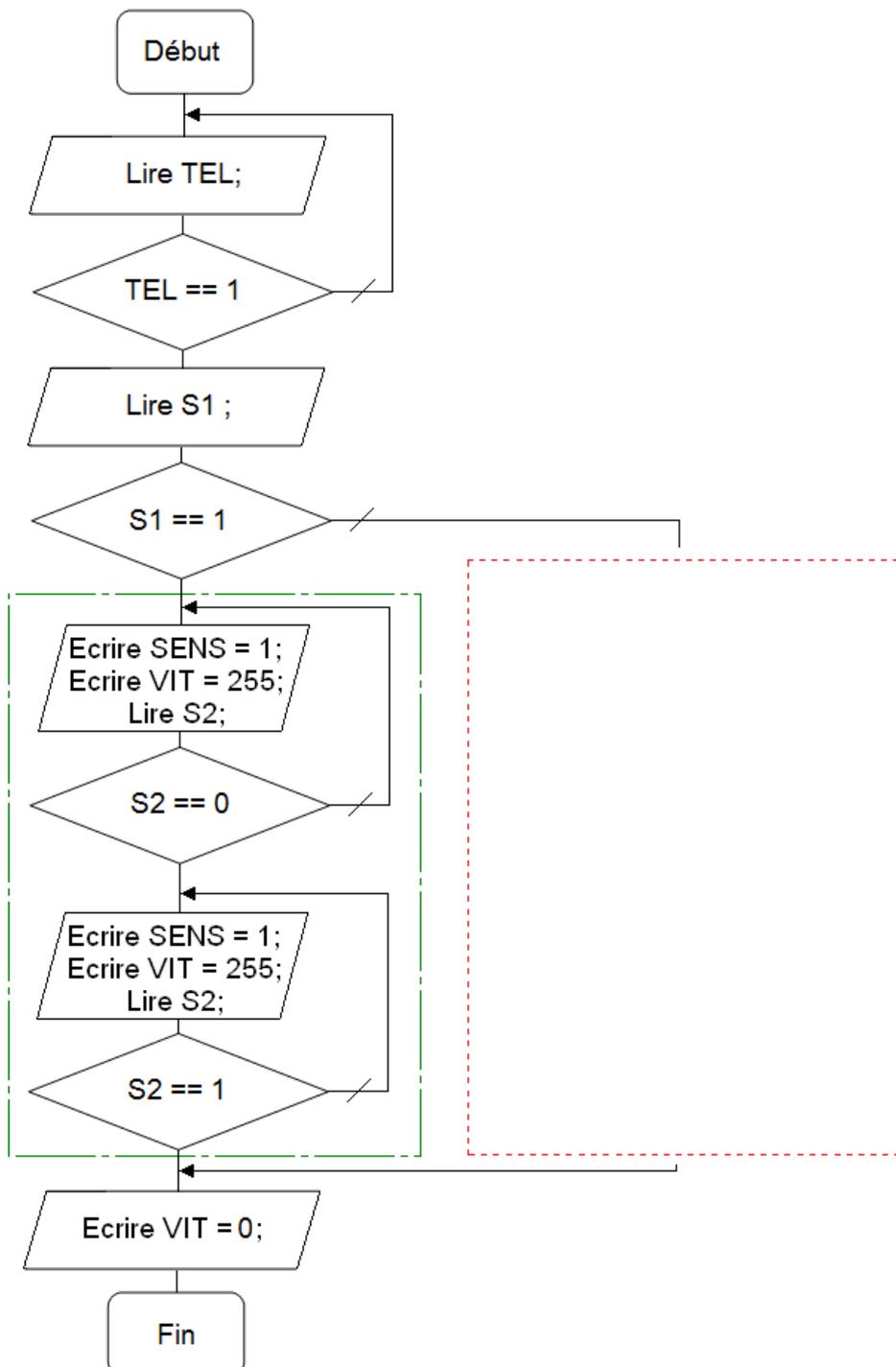
**Q4 / COMPLÉTER** les chronogrammes (variables SENS, VIT). On suppose que les mouvements s'effectuent à vitesse maximum.



**Q5 / EXPLIQUER** à quel mouvement de l'arceau correspond la partie d'algorithme entourée par le trait mixte vert.

**Q6** / Toujours dans ce cadre, **EXPLIQUER** pourquoi il est nécessaire d'effectuer le test  $S2==0$ .

**Q7** / Dans le cadre en pointillés rouges, **ETABLIR** le traitement correspondant à l'autre mouvement.



Algorithme

## 2 / Pilotage évènementiel du système – Fonctionnement de base

**OUVRI**R le modèle *Matlab* « Vigipark.mdl ».

**OUVRI**R, par double clic, le diagramme d'états qui permet d'accéder à l'éditeur spécialisé *Stateflow*.

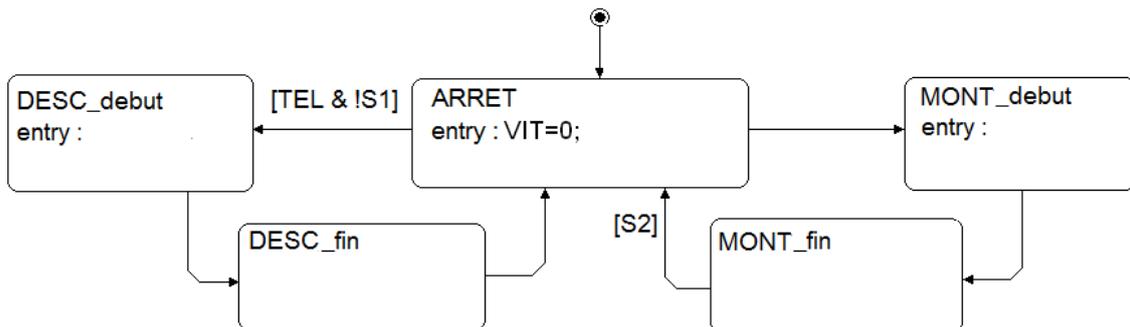
Remarques :

- Toutes les variables nécessaires ont déjà été créées (**RESPECTER** majuscules et minuscules).
- **ENREGISTRER** régulièrement le fichier modèle.

**Q8 / COMPLÉTER** les transitions et les actions manquantes.

**TESTER** le fonctionnement, **FAIRE VALIDER**. □

**COMPLÉTER** le diagramme états-transitions ci-dessous.



**Q9 / REGROUPER** les états correspondant à la montée dans un super-état MONTER : pour cela, **CRÉER** d'abord le super-état autour des états MONT\_debut, MONT\_fin puis **CLIQUER DROIT** et **SÉLECTIONNER** la commande : *Make contents/subcharted*. L'accès aux sous-états s'effectue ensuite par double clic sur le super-état.

**EFFECTUER** la même manipulation pour les états correspondants à la descente (super-état DESCENDRE).

**REPRODUIRE** le diagramme états-transitions ainsi constitué ci-dessous.

## 4 / Prise en compte de la présence d'un véhicule

On souhaite maintenant améliorer le fonctionnement en prenant en compte la présence d'un véhicule. La variable correspondante est « VEHICULE ».

**Q10 / EXPLIQUER** quel est le mouvement à interdire lorsque cette variable est au NL « 1 ». **MODIFIER** en conséquence une transition dans le diagramme états-transitions.  
**TESTER** le fonctionnement, **FAIRE VALIDER**.   
**REPRODUIRE** la transition sur le diagramme de la question 9, **FAIRE APPARAÎTRE** les modifications en **VERT**.

On dispose de trois variables de sorties (Haut, Bas, Move) encore inutilisées, permettant d'émettre des comptes-rendus sur l'état de l'arceau.

Remarque : on suivra l'évolution de ces variables sur les *displays* correspondants du modèle *Matlab*.

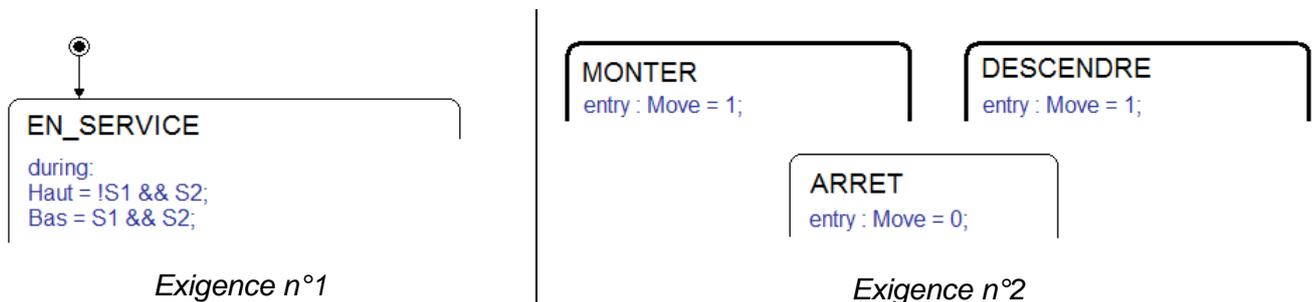
Exigences n° :

1. Haut et Bas sont au NL « 1 » respectivement quand l'arceau est en fin de course haute / basse.
2. Move est au NL « 1 » quand l'arceau est en mouvement.

**Q11 / PROPOSER** puis **METTRE EN ŒUVRE** une solution permettant de satisfaire les exigences énoncées ci-dessus.

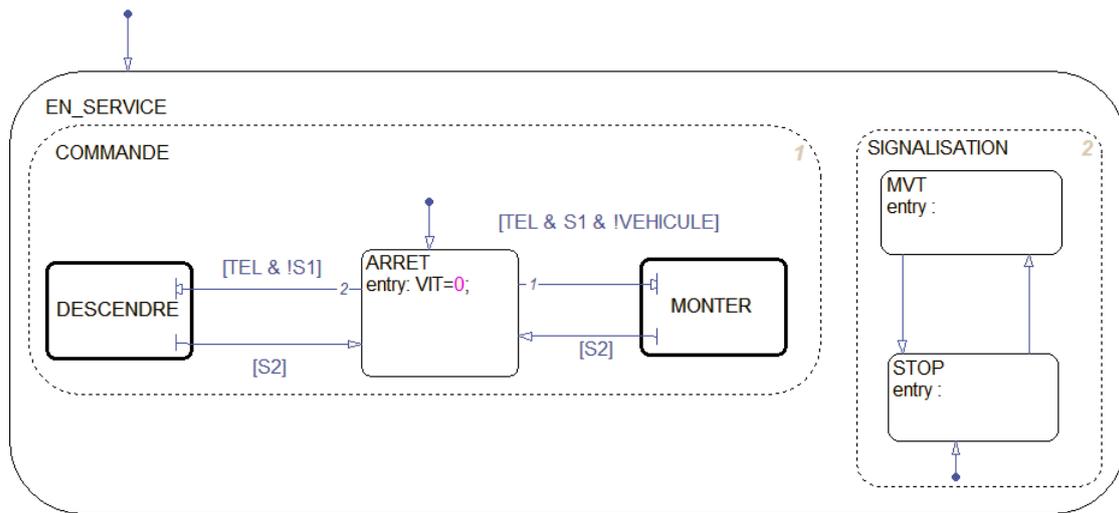
**TESTER** le fonctionnement, **FAIRE VALIDER**.

**FAIRE APPARAÎTRE** les modifications ci-dessous, en **BLEU**.



On opte pour une autre solution pour satisfaire l'exigence relative à la variable MOVE : **RAJOUTER** l'état **SIGNALISATION**, conformément à la figure ci-dessous. Après sélection de l'état **EN\_SERVICE**, **CLIQUER DROIT** et **OPTER** pour la commande *Decomposition/Parallel (AND)*.

Les états **COMMANDE** et **SIGNALISATION** apparaissent alors avec un trait en pointillés et sont affectés des numéros 1, 2 : ils sont simultanément actifs.



**Q12 / RENSEIGNER** le rectangle état SIGNALISATION.

**TESTER** le fonctionnement, **FAIRE VALIDER**. □

**REPRODUIRE** ci-dessous le rectangle état SIGNALISATION.

Remarque : l'arceau est en mouvement quand la vitesse est différente de zéro, il est à l'arrêt quand la vitesse est nulle.

