

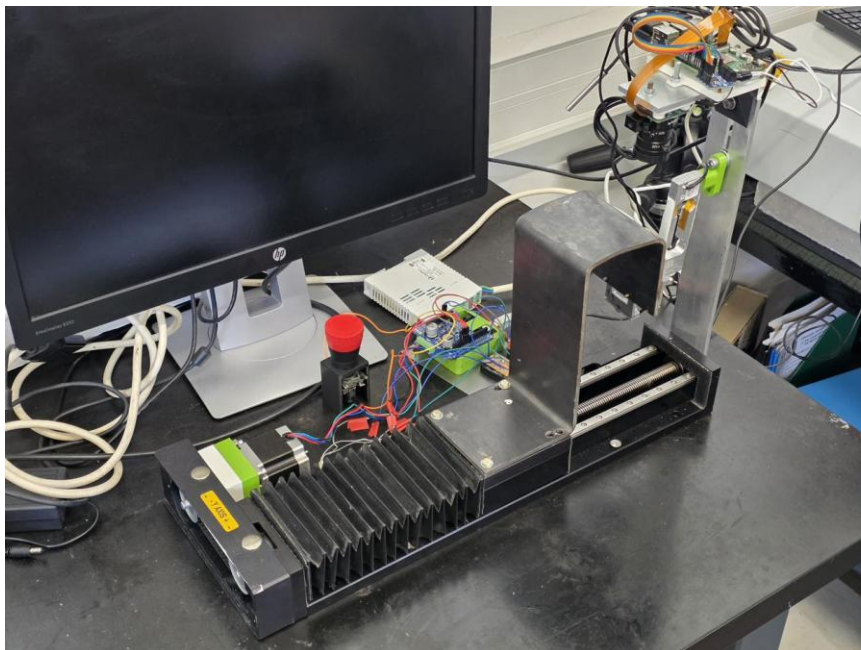
RAPPORT DE STAGE

BUT Génie Mécanique et Productique — 2^e année

Parcours Conception de Produits Durables (CPD) — Année 2025/2026

Caractérisation mécanique d'un solide cellulaire pressurisé : le parenchyme de pomme de terre

*Mesure du module de Young et estimation du coefficient de Poisson
en fonction de la concentration osmotique*



Sujet : « Mechanical Characterization of Pressurized Cellular Solid: Potato Parenchyma »

Imran BIN MOHD ZAKI

Soutenance : 22 juin 2026

Information	Détail
Établissement	IUT d'Aix-Marseille — Département Génie Mécanique et Productique, 413 av. Gaston Berger, 13625 Aix-en-Provence
Laboratoire d'accueil	Institut des Sciences du Mouvement (ISM, UMR 7287) — équipe Systèmes Bio-Inspirés (SBI)
Lieu du stage	Halle d'essai de l'IUT d'Aix-en-Provence
Période	Du 13/04/2026 au 19/06/2026 (300 h de présence effective)
Tuteur entreprise	M. Loïc TADRIST — enseignant-chercheur
Tuteur universitaire	M. Eric JEANCELME

Fiche de confidentialité

FICHE D'IDENTITE ET DE CONFIDENTIALITE	
Année 2026	
STAGIAIRE : Imran BIN MOHD ZAKI	
TITRE : Mechanical Characterization of Pressurized Cellular Solid: Potato Parenchyma	
RESUME : Conception, fabrication et instrumentation d'un nouveau banc de traction (C-satia) vs. impression 3D, usinage, usquithao Raepberry Pi (Nadl: REO) pour caractériser le parenchyme de pomme de terre, modèle de solide cellulaire pressurisé. Mesure du Module de Young et estimation du coefficient du Poisson.	
Nombre de pages : 33 Nombre de références bibliographiques : 7	
ENTREPRISE : Aix-Marseille Université	
Président directeur général : Eric BERION	
Nombre d'employés : 8000 personnes	
Domaine d'activité : Education et recherche	
Adresse :	
Téléphone : Télécopie :	
Tuteur industriel : Loïc TADRIST	
Fonction : Enseignant chercheur	
Mail : loic.tadrist@univ-amu.fr Téléphone : +33413946393	
PARTIE A REMPLIR PAR LE TUTEUR INDUSTRIEL	
Accessibilité de ce rapport (entourer la mention correcte) :	
LIBRE	CONFIDENTIEL pendant an(s)
Date : 16/06/2026	Nom du stagiaire : BIN MOHD ZAKI Signature :

Figure 1 — Fiche de confidentialité.

Remerciements

Je remercie sincèrement M. Loïc TADRIST, mon tuteur de stage dans l'équipe Systèmes Bio-Inspirés de l'ISM. Il m'a accueilli sur ce projet, m'a fait confiance, m'a conseillé tout au long du stage et m'a aidé à usiner le trou oblong du support de la cellule de charge.

Je remercie tout particulièrement M. Louison FIORE pour son aide et son expérience. Lors de ses travaux précédents, il avait encadré Rayan CHAIX. Il m'a donné les fichiers Arduino qui pilotent le moteur et des exemples de code pour la carte IHM03A1, qui m'ont beaucoup aidé pendant le débogage, ainsi que les exemples pour faire fonctionner la caméra. Il m'a aussi donné son avis sur la conception des mors et du support de la caméra et du Raspberry Pi, et il a fabriqué les moules à glace utilisés pour les essais.

Je remercie M. Eric JEANCELME pour son aide pendant la fabrication de l'extension de l'arbre du moteur (tournage CNC et conventionnel).

Je remercie M. Noé SIONNEAU pour sa démonstration de l'impression 3D en résine (Formlabs, PreForm, Form3L).

Enfin, je remercie chaleureusement mes amis malaisiens pour leur bonne humeur pendant le stage et pour leurs conseils lors de la fabrication de certaines pièces.

Sommaire

Fiche de confidentialité.....	2
Remerciements.....	3
Sommaire	4
Glossaire	5
1. Introduction.....	6
2. Cadre du stage.....	6
2.1. L'Institut des Sciences du Mouvement (ISM).....	6
2.2. L'équipe Systèmes Bio-Inspirés (SBI).....	7
2.3. La halle d'essai de l'IUT d'Aix-en-Provence.....	8
2.4. Continuité avec le stage précédent.....	8
3. Rapport technique	8
3.1. Le sujet de la mission.....	8
3.2. Démarches prévisionnelles et réalisées.....	11
3.3. Développement des phases de la mission	11
3.4. Bilan technique et perspectives.....	31
4. Rapport PPP.....	31
4.1. Description d'un métier : doctorant en biomécanique (ISM).....	31
4.2. Activités annexes	32
5. Conclusion générale et personnelle	32
Bibliographie et webographie	33
Annexes.....	34
Annexe A — Données par échantillon	34
Annexe B — Dessins de définition (CATIA V5).....	35
Annexe C — Programme Arduino (firmware du banc).....	44
Annexe D — Flux Node-RED.....	49
Annexe E — Scripts MATLAB de traitement des données	50
4 ^e de couverture.....	58
Résumé.....	58

Glossaire

Terme / Sigle	Définition
Parenchyme	Tissu végétal fait de cellules à parois fines, remplies d'eau ; c'est le principal constituant de la chair de la pomme de terre.
Turgescence	Tension d'une cellule végétale due à la pression de l'eau contre la paroi de la cellule.
Pression osmotique	Pression liée à la différence de concentration en solutés de part et d'autre d'une membrane ; elle commande les échanges d'eau de la cellule.
Module de Young (E)	Grandeur (en MPa) qui mesure la rigidité d'un matériau : rapport entre la contrainte et la déformation.
Coefficient de Poisson (ν)	Rapport entre la déformation transversale et la déformation longitudinale en traction.
Sorbitol	Sucre-alcool ($M = 182,17 \text{ g/mol}$) utilisé comme agent osmotique pour faire varier la turgescence.
Palier	Avance imposée (0,1 mm) suivie d'un temps d'attente (90 s) qui laisse l'échantillon se relâcher.
Perte de course	Partie du déplacement du moteur qui n'est pas transmise à l'échantillon (glissement aux mors et flexion de la cellule).
HX711	Convertisseur 24 bits qui lit la cellule de charge à jauges.
Node-RED	Logiciel de programmation par blocs utilisé pour piloter le banc et enregistrer les données.
CAO	Conception Assistée par Ordinateur (logiciel CATIA V5 dans ce projet).

1. Introduction

Dans le cadre de la deuxième année de BUT Génie Mécanique et Productique (parcours Conception de Produits Durables) à l'IUT d'Aix-Marseille, j'ai réalisé un stage à l'Institut des Sciences du Mouvement (ISM), dans l'équipe Systèmes Bio-Inspirés (SBI), à la halle d'essai de l'IUT d'Aix-en-Provence, du 13 avril au 19 juin 2026.

Ce stage fait partie d'un projet de recherche qui étudie le comportement mécanique du parenchyme de pomme de terre. Ce tissu sert de modèle d'un solide cellulaire pressurisé : sa rigidité dépend surtout de la pression de turgescence à l'intérieur des cellules. Comprendre comment cette rigidité change quand on modifie l'état hydrique du tissu est utile pour concevoir des structures et des actionneurs souples inspirés du vivant (« muscles végétaux »).

Le banc d'essai servant à ces mesures était déjà utilisé par un autre stagiaire. Ma mission a donc été de concevoir et de fabriquer un nouveau banc de traction, semblable au précédent mais plus fiable, puis de mener une campagne de mesures complète. Le but est de mesurer le module de Young — et, si possible, le coefficient de Poisson — du parenchyme en fonction de la concentration de la solution de trempage (sorbitol, de 0 à 0,5 mol/L).

Ce rapport présente d'abord le cadre du stage (l'ISM, l'équipe SBI et la halle d'essai). Il décrit ensuite le travail technique : le sujet, la démarche, la conception et la fabrication du banc, le protocole, l'acquisition et le traitement des données, les résultats et la discussion. Il se termine par un rapport PPP et une conclusion personnelle.

2. Cadre du stage

2.1. L'Institut des Sciences du Mouvement (ISM)

L'Institut des Sciences du Mouvement Étienne-Jules Marey est une unité mixte de recherche (UMR 7287) liée au CNRS et à Aix-Marseille Université. Il étudie le mouvement humain et animal avec des approches expérimentales, théoriques et technologiques, de la biomécanique et de l'ingénierie jusqu'aux neurosciences. L'institut comprend trois équipes : BioMécanique/bioIngénierie (BMI), Dynamiques Comportementales & Cognition (DynamICC) et Systèmes Bio-Inspirés (SBI).



Figure 2 — Localisation de l'ISM (Parc scientifique de Luminy).

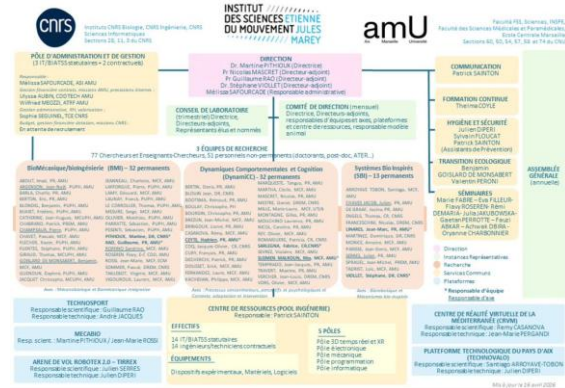


Figure 3 — Organigramme de l'ISM (trois équipes).

2.2. L'équipe Systèmes Bio-Inspirés (SBI)

L'équipe SBI s'inspire du vivant pour concevoir des solutions techniques nouvelles, dans une logique de sobriété et de durabilité. Ses travaux se partagent en deux axes : un axe « Biorobotique » (robots et capteurs inspirés du vivant) et un axe « Mécanismes bio-inspirés » (étude de structures biologiques pour en tirer des principes de conception). Mon projet appartient à ce second axe, avec l'étude des propriétés mécaniques du parenchyme de pomme de terre.

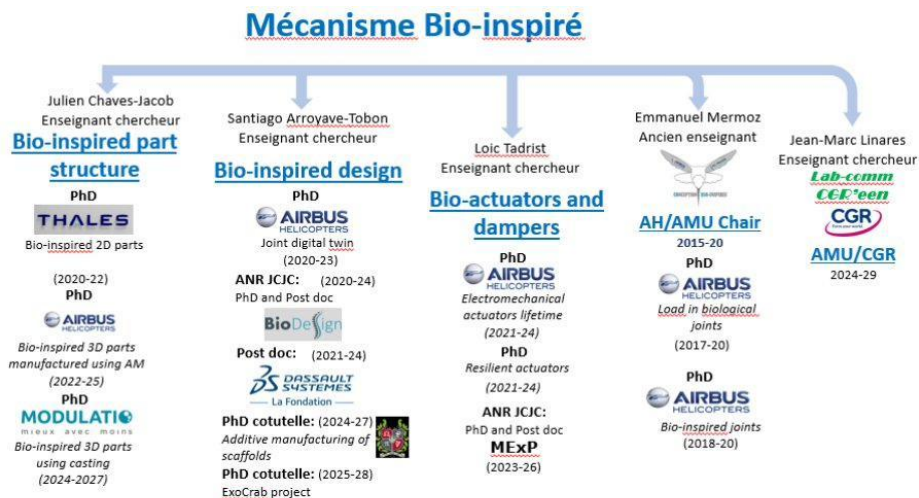


Figure 4 — Organisation de l'équipe Systèmes Bio-Inspirés (SBI).

2.3. La halle d'essai de l'IUT d'Aix-en-Provence

Le stage s'est déroulé à la halle d'essai de l'IUT d'Aix-en-Provence. C'est un espace technique dédié aux essais, équipé en particulier d'imprimantes 3D (dépôt de fil et résine). J'ai aussi eu accès à l'atelier du département GMP pour les opérations d'usinage (perçage, fraisage, tournage).

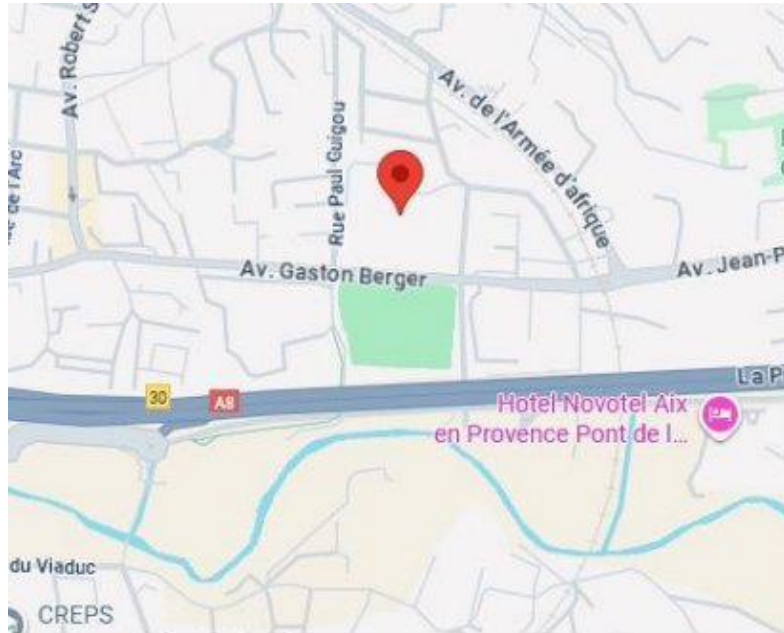


Figure 5 — Localisation de la halle d'essai de l'IUT d'Aix-en-Provence.

2.4. Continuité avec le stage précédent

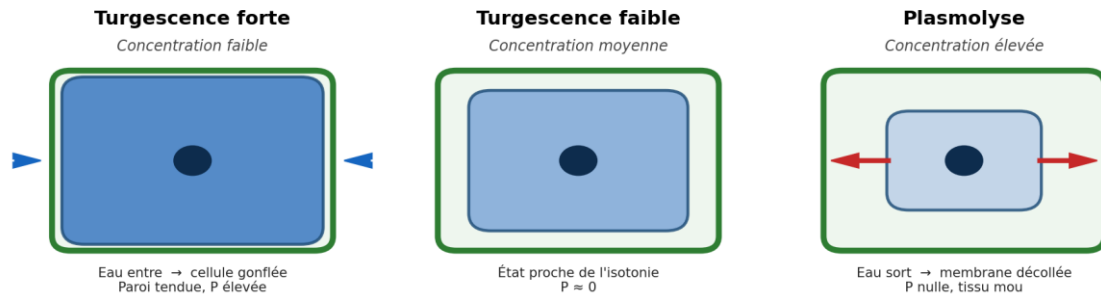
Le banc de référence avait été reconçu par Rayan CHAIX (2025), qui avait renforcé le bâti et conçu de nouveaux mors imprimés en 3D. À la fin de son stage, le module de Young avait été estimé pour les concentrations 0,1 à 0,3 mol/L, mais le coefficient de Poisson n'avait pas été donné et le glissement aux mors restait un problème. Mon stage reprend cette base. Comme le banc existant était déjà utilisé, j'ai dû en construire un nouveau, semblable, puis réaliser la campagne complète sur la série 0–0,5 mol/L.

3. Rapport technique

3.1. Le sujet de la mission

3.1.1. Contexte et problématique

Le parenchyme de pomme de terre est un solide cellulaire pressurisé. Sa rigidité dépend surtout de la pression de turgescence, c'est-à-dire la pression de l'eau qui gonfle les cellules contre leurs parois. On contrôle cette turgescence par osmose : quand on plonge les échantillons dans des solutions de plus en plus concentrées, l'eau sort des cellules, la turgescence baisse et le tissu devient plus mou. L'équipe SBI veut mesurer cet effet, car il aide à comprendre les « muscles végétaux » qui inspirent un projet d'actionneur souple.



Concentration osmotique croissante → turgescence et rigidité décroissantes

Figure 6 — Effet de la concentration osmotique sur une cellule du parenchyme : à gauche l'eau entre et la cellule est turgescence (tissu rigide) ; à droite l'eau sort et la cellule se plasmolyse (tissu mou).

La mission comprend deux livrables : (i) la courbe du module de Young apparent E en fonction de la concentration en sorbitol (0 à 0,5 mol/L) et (ii) une estimation du coefficient de Poisson ν sur la même plage. Le but n'est pas de mesurer la rupture, mais le comportement élastique du tissu selon son état de turgescence.

3.1.2. État de l'art (synthèse bibliographique)

Avant de choisir une solution, j'ai comparé les dispositifs qui servent à mesurer les propriétés mécaniques du parenchyme. Le tableau suivant résume cette synthèse.

Dispositif	Description	Référence
Banc d'essai mécanique existant	Banc conçu à l'IUT par un étudiant de 3 ^e année : tôle pliée, mâchoires en résine et système d'acquisition optique. Mesure E et ν ; c'est la base de mon projet.	Chaix R. (2025)
Banc numérique (commercial)	Essais de traction et de compression sur deux variétés (Spunta, Daïfla) avec une machine de traction numérique du commerce. Mesure E et ν , mais coût élevé et non fabriquée au laboratoire.	ResearchGate
AFM + théorie de coque élastique	Indentation par microscopie à force atomique sur cellules végétales (épiderme d'oignon) avec simulations par éléments finis. Sépare le module de la paroi et la pression de turgescence, mais ne donne pas le ν macroscopique.	ResearchGate
Compression + DIC sur tiges	Mesure de E et ν en compression axiale sur des tiges de pomme de terre, avec une caméra IMX415 (corrélation d'images) et une machine universelle.	MDPI 2022
DIC sur tissus végétaux	Corrélation d'images 2D et 3D sur des tissus végétaux mous (gui, cactus, feuilles) : traction uniaxiale et biaxiale pour mesurer les champs de déformation.	PMC

Cette comparaison montre que les solutions du commerce sont précises mais chères, et que l'AFM ne donne pas le coefficient de Poisson global. La voie la plus adaptée au stage est donc de reprendre et d'améliorer le banc optique de l'IUT.

3.1.3. Choix de la solution technique

Les bancs de traction du commerce qui mesurent en millinewtons (Instron, Linkam, ADMET) sont trop chers (15 000 à 80 000 €) et trop longs à obtenir pour la durée du stage. La microscopie

à force atomique (AFM) ne mesure que la paroi de la cellule et ne donne pas le coefficient de Poisson global. La solution choisie a donc été de concevoir et de fabriquer un banc d'essai dédié, inspiré du banc existant et instrumenté (cellule de charge, acquisition par Raspberry Pi, mors imprimés). C'est la solution la plus économique et la mieux adaptée au stage.

3.1.4. Analyse fonctionnelle

Avant de rédiger le cahier des charges, j'ai réalisé une analyse fonctionnelle du banc. La bête à cornes précise le besoin : à qui le banc rend service, sur quoi il agit et dans quel but.

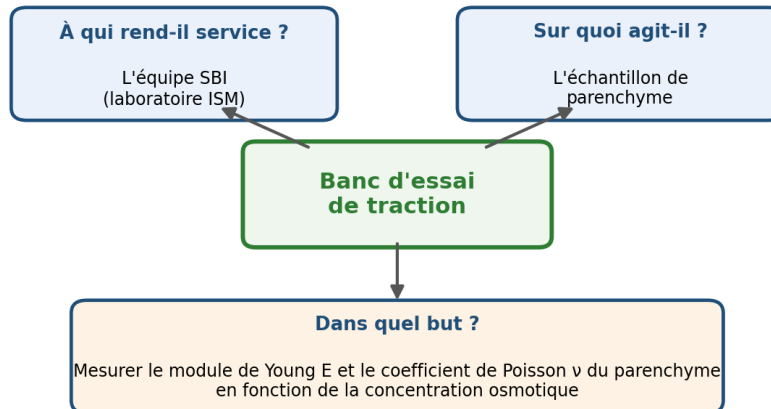


Figure 7 — Bête à cornes du banc d'essai.

Les fonctions du banc (fonction principale et fonctions de contrainte) qui découlent de cette analyse sont détaillées et chiffrées dans le cahier des charges fonctionnel ci-dessous.

3.1.5. Cahier des charges fonctionnel

Le tableau ci-dessous donne les fonctions et les contraintes du banc. La colonne « Niveau » indique le degré d'obligation : F0 = impératif, F1 = important (peu négociable), F2 = négociable.

Réf.	Fonction / Contrainte	Critère	Niveau
FP1	Maintenir l'échantillon	Tenir des cylindres $\varnothing \approx 5$ mm sans glissement	F0
FP2	Appliquer une traction uniaxiale	Avance contrôlée de 0,1 mm par palier	F0
FP3	Immerger en solution à froid	Essai en eau + sorbitol, bain de glace 2–8 °C	F0
FP4	Mesurer la force	Cellule de charge 300 g, sortie numérique CSV	F0
FP5	Suivre la géométrie	Une image par palier (allongement, diamètre)	F1
FP6	Acquérir et exporter	Données en .csv + images horodatées	F1
FC1	Compatibilité système	S'adapter au guidage linéaire et au moteur existants	F0
FC2	Rigidité des supports	Limiter les jeux et les déformations parasites	F1

Réf.	Fonction / Contrainte	Critère	Niveau
FC3	Sécurité	Arrêt d'urgence matériel sur l'alimentation moteur	F0
FC4	Matériaux	Matériaux non réactifs à l'eau et au sorbitol	F1
FC5	Réglage	Réglage en hauteur de la cellule de charge	F2

3.2. Démarches prévisionnelles et réalisées

Le plan prévu était : une courte prise en main, la conception et la fabrication du banc, puis la campagne de mesures et le traitement des données. En réalité, la fabrication a pris plus de temps que prévu. La mise au point des mors a été un travail continu : j'ai dû tester plusieurs versions pour trouver un mors capable de tenir l'échantillon sans glissement. De plus, le débogage du moteur a duré plus longtemps que prévu, car le réglage de sa configuration était complexe. La campagne de mesures et le traitement (avec une nouvelle méthode de mesure des déformations) ont occupé la dernière partie du stage.

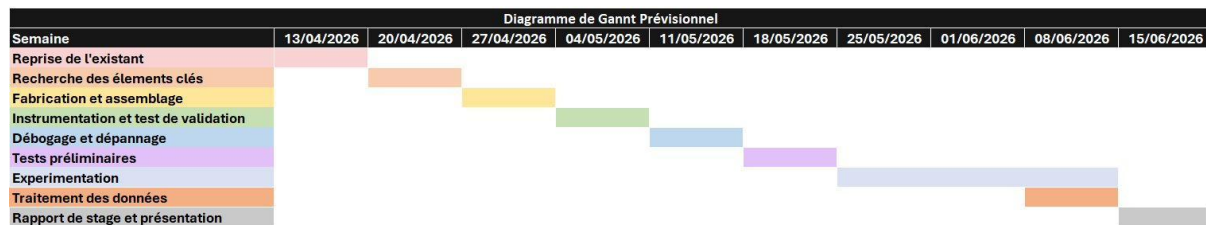


Figure 8 — Diagramme de Gantt prévisionnel.

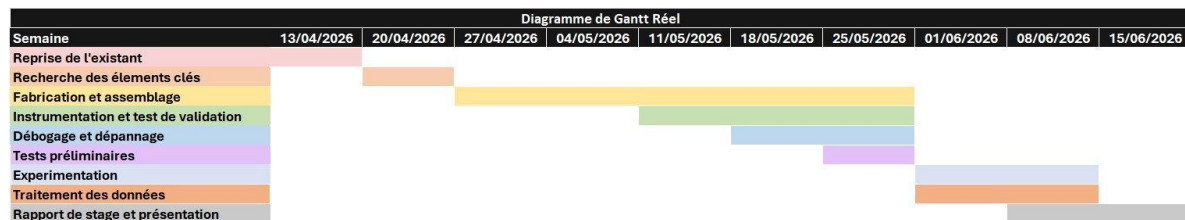


Figure 9 — Diagramme de Gantt réel. La phase de fabrication des mors et le débogage du moteur sont plus longs que prévu.

3.3. Développement des phases de la mission

3.3.1. Analyse du dispositif existant

Le banc de référence ne pouvait pas être utilisé, car un autre stagiaire s'en servait. On m'a donc fourni un squelette de guidage linéaire identique au sien comme point de départ. Dans ce système, le moteur transmet sa rotation par une courroie à une vis, qui déplace un chariot le long du guide. Ce chariot portait déjà les capteurs de fin de course. À partir de cette base, je devais intégrer un moteur, construire la mesure de force et l'acquisition optique, et concevoir les mors.



Figure 10 — Squelette du guidage linéaire fourni (moteur, courroie, vis et chariot portant les capteurs de fin de course).

3.3.2. Conception du banc (CAO)

La figure ci-dessous présente le schéma de principe du banc. L'échantillon est immergé dans le bac (eau + soluté osmotique) et tenu par deux mors : l'un est relié à la cellule de charge, l'autre est entraîné par le moteur NEMA 23 à travers une courroie et une vis. En déplaçant le chariot, le moteur étire l'échantillon : la cellule de charge, lue par le module HX711, mesure la force, pendant que la caméra HQ enregistre l'allongement. Le Raspberry Pi 5 assure l'acquisition (force, images et température via la sonde), tandis que l'Arduino et le driver pilotent le moteur. Ce schéma situe les principaux composants avant la présentation détaillée de la conception.

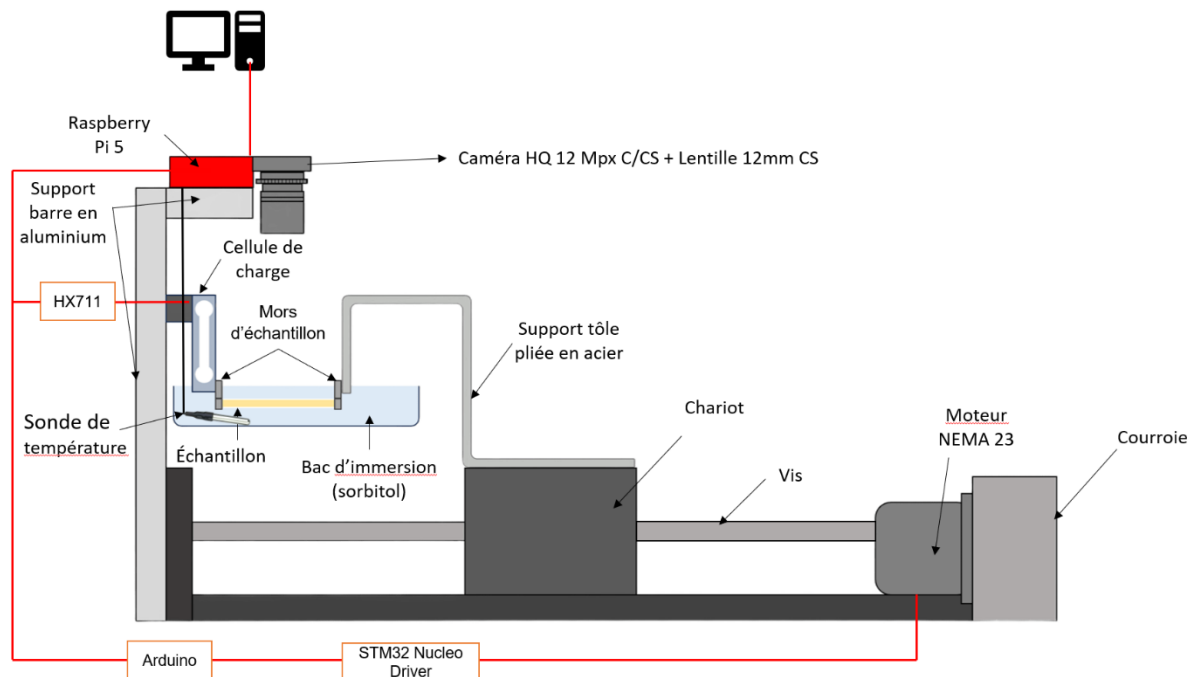


Figure 11 — Schéma de principe

J'ai conçu l'ensemble sous CATIA V5. Le banc reprend le guidage linéaire à courroie et un moteur pas-à-pas NEMA 23, comme le banc précédent. La traction s'applique entre un mors fixé sur le chariot mobile et un mors fixé sur la cellule de charge, du côté fixe. L'architecture repose sur deux plaques en aluminium : la première plaque, fixée au guidage linéaire, porte la cellule de charge ; la seconde plaque, fixée à la première par une équerre (vis et écrous M5), porte la caméra. Plusieurs pièces d'adaptation relient ces sous-ensembles.

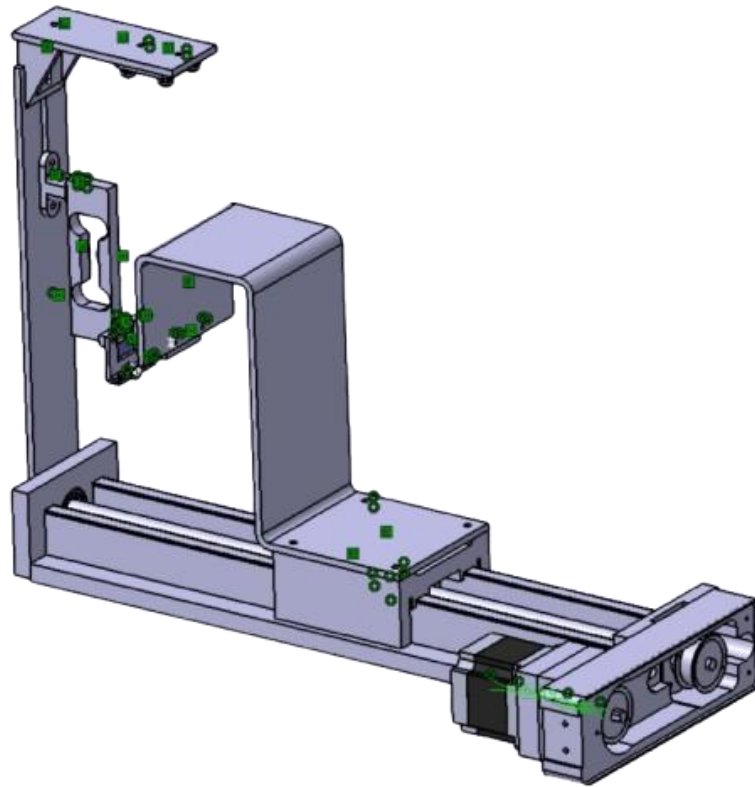


Figure 12 — Vue CAO de l'assemblage du banc (CATIA V5).

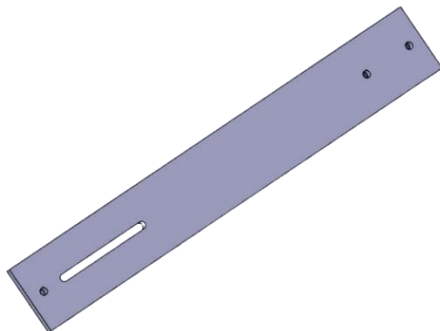


Figure 13 — Plaque support (alu) de la cellule de charge.

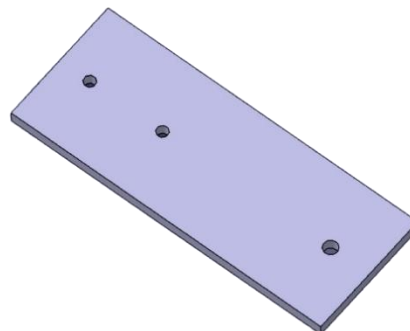


Figure 14 — Plaque support (alu) de la caméra.

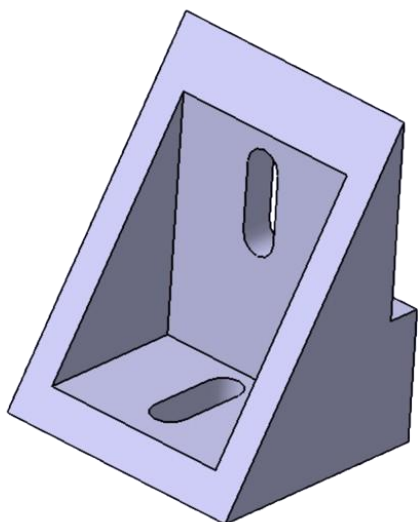


Figure 15 — Équerre reliant les deux plaques.

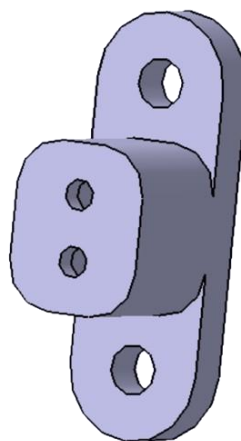


Figure 16 — Pièce d'espacement de la cellule de charge.

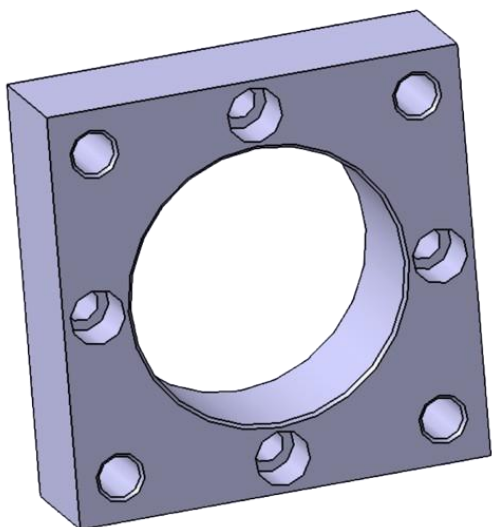


Figure 17 — Pièce d'adaptation primaire.

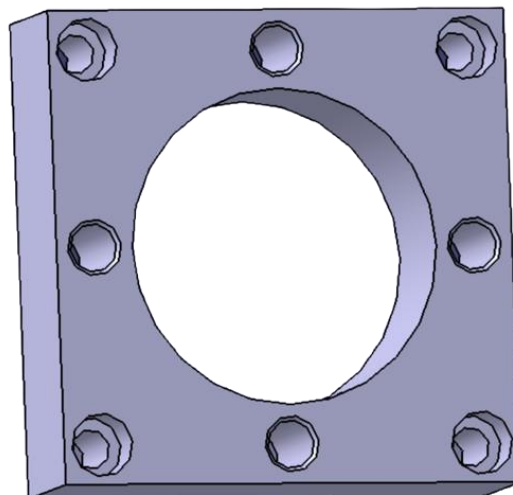


Figure 18 — Pièce d'adaptation secondaire.

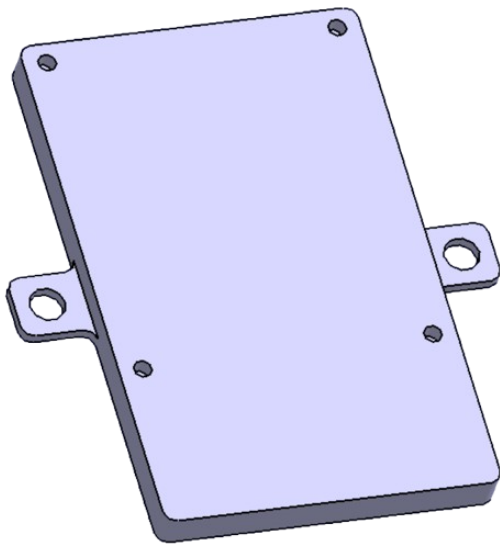


Figure 19 — Logement du Raspberry Pi 5.

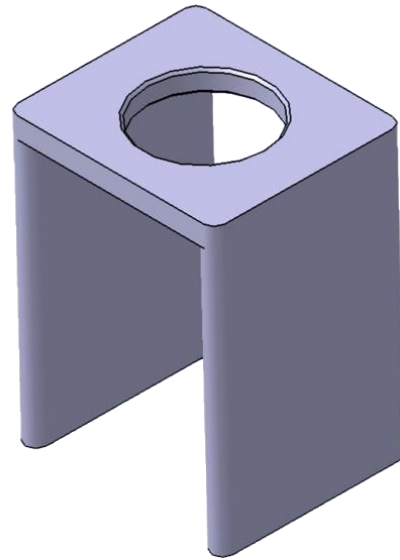


Figure 20 — Monture de l'arrêt d'urgence.

Le mors de Rayan, conçu en 2025, serrait l'échantillon entre deux mâchoires imprimées en 3D maintenues fermées par des pinces de bureau. Ce principe est simple et rapide à monter, et c'est avec lui qu'avaient été obtenus les premiers modules de Young du parenchyme ; je l'ai donc repris comme point de départ, en arrondissant les bords de l'ouverture de la mâchoire pour réduire la concentration de contrainte.

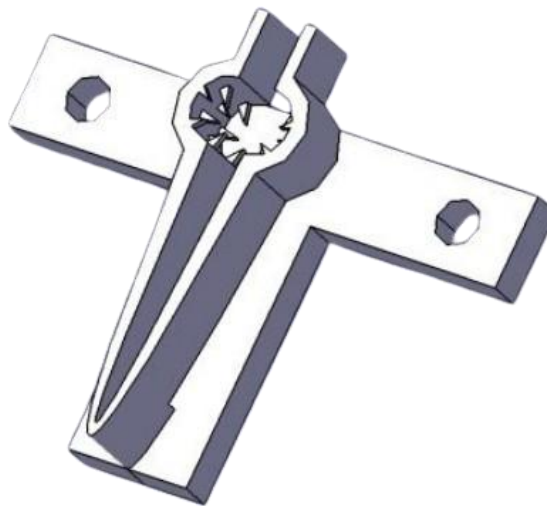


Figure 21 — Mors de Rayan

La mise au point des mors a été la phase la plus longue : c'est un travail itératif où chaque version est testée puis modifiée jusqu'à obtenir un maintien fiable. J'ai conçu et comparé trois mors : un mors « SnapFit » (encliquetage), une reprise du mors de Rayan (serrage par pinces) et un mors « vis-écrou ».

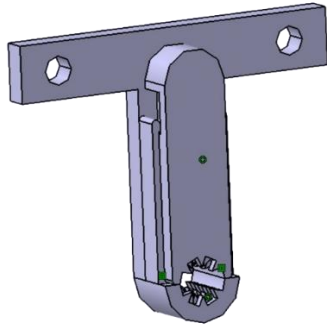


Figure 22 — Mors SnapFit (encliquetage).

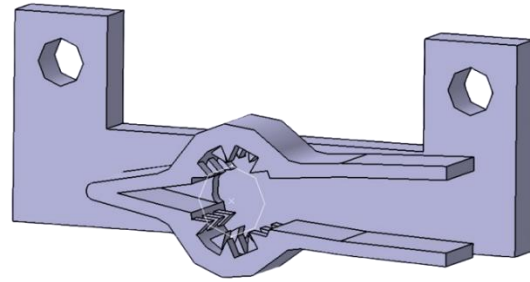


Figure 23 — Reprise du mors de Rayan (serrage par pinces).

La reprise du mors de Rayan est fragile à cause des fortes contraintes sur les bords autour de l'ouverture de la mâchoire (zones mises en évidence sur la figure ci-dessous). Même si ces bords sont arrondis pour réduire la concentration de contrainte, la pièce peut encore casser sous l'effort de serrage de la pince ; de plus, la pince se détache sans cesse de la zone où elle doit se fixer, ce qui rend le montage difficile et répartit mal l'effort. Le mors SnapFit, lui, glisse : la force de serrage d'un encliquetage n'est pas assez forte pour bien tenir l'échantillon.

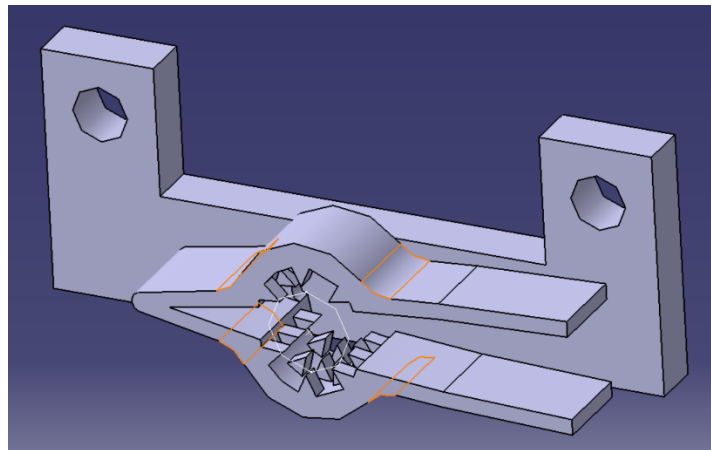


Figure 24 — Reprise du mors de Rayan : les bords autour de l'ouverture de la mâchoire (en évidence) subissent de fortes contraintes et peuvent rompre sous le serrage de la pince.

Le troisième concept, le mors vis-écrou, serre l'échantillon entre deux pièces à l'aide d'une vis M3 et d'un écrou M3. Les trois mors sont imprimés dans la même résine (Grey V4, Formlabs), donc la tenue à l'immersion ne les distingue pas. Le choix final entre ces trois concepts est présenté dans la partie fabrication (3.3.4).

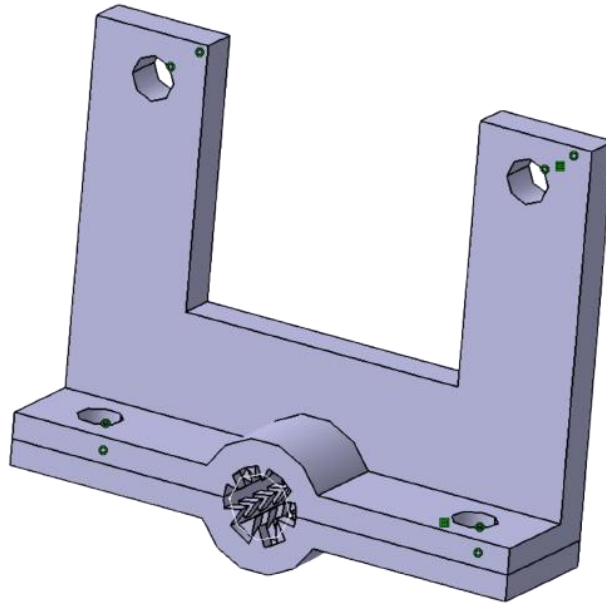


Figure 25 — Mors vis-écrou (concept retenu).

3.3.3. Choix du matériau des mors

Le matériau des mors demande une attention particulière, car ce sont les pièces qui tiennent l'échantillon et qui sont en contact avec l'eau pendant l'essai. Le PLA, soumis à l'immersion, subit une plastification : les molécules d'eau s'insèrent entre les longues chaînes du polymère, et le matériau devient souple, caoutchouteux et mou ; il ne peut donc plus tenir l'échantillon de façon fiable. Le TPU a été écarté car il est trop souple et trop flexible : il ne serre pas bien l'échantillon. Les mors sont donc imprimés en résine Grey V4 (Formlabs), plus rigide et précise, et qui ne se ramollit pas dans l'eau.

Matériau	Comportement pour les mors	Choix
PLA	Se plastifie dans l'eau : devient souple et mou	Écarté
TPU	Trop souple et flexible : ne serre pas l'échantillon	Écarté
Résine Grey V4	Rigide, précise et stable dans l'eau	Retenu

3.3.4. Fabrication des pièces

La fabrication a utilisé trois procédés. Les mors ont été imprimés en résine Grey V4 sur une Formlabs Form3L, puis nettoyés à l'alcool et durcis aux UV pour atteindre leurs propriétés finales. Les pièces d'adaptation et de montage (équerre, pièces d'adaptation primaire et secondaire, monture de la caméra, pièce d'espacement de la cellule de charge, logement du Raspberry Pi, logement de l'Arduino et monture de l'arrêt d'urgence) ont été imprimées en PLA. Elles reçoivent des inserts M4 et s'assemblent par vis M4 ; la pièce d'espacement de la cellule de charge est tenue par une vis et un écrou M6.

Les pièces en aluminium ont été usinées : les deux plaques support et l'extension de l'arbre du moteur. L'extension a été réalisée par tournage à commande numérique (SOMAB 250+) puis par tournage conventionnel (RAMO), avec un ajustement ajusté pour entrer dans l'arbre du moteur, sous la conduite de M. Eric JEANCELME. Deux vis sans tête permettent de bloquer

l'extension sur l'arbre du moteur. Un trou oblong a aussi été usiné par tournage conventionnel dans le support de la cellule de charge, pour permettre un réglage en hauteur, sous la conduite de M. Loïc TADRIST. Les dessins de définition des pièces sont donnés en annexe.



Figure 26 — Pièce d'adaptation primaire imprimée en PLA.

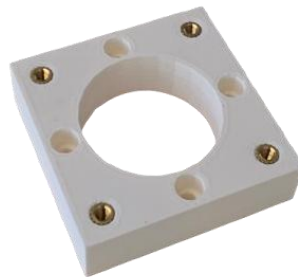


Figure 27 — Pièce d'adaptation secondaire imprimée en PLA (inserts M4 laitonnés).



Figure 28 — Extension de l'arbre du moteur, usinée (tournage CNC puis conventionnel).



Figure 29 — Support aluminium usiné, cellule de charge et pièce d'adaptation.



Figure 30 — Mors SnapFit imprimé en résine.



Figure 31 — Reprise du mors de Rayan imprimée en résine.



Figure 32 — Mors vis-écrou imprimé en résine (concept retenu).

Parmi les trois concepts de mors conçus, le tableau suivant compare leurs performances (note sur 5 ; une note élevée est meilleure). Le mors vis-écrou, qui offre le meilleur maintien tout en restant simple à monter et robuste, a été retenu et fabriqué en résine.

Critère	SnapFit	Reprise Rayan	Vis-écrou	Observation
Maintien (anti-glissement)	3	2	4,5	Vis-écrou : meilleur serrage
Facilité de montage	4	2	4	Pincers de Rayan peu pratiques
Robustesse	5	1,5	5	Reprise Rayan très fragile
Retenu	—	—	✓	Mors vis-écrou

Les deux plaques support en aluminium, qui portent la cellule de charge et la caméra, sont montrées ci-dessous.



Figure 33 — Plaque support en aluminium (caméra), usinée.



Figure 34 — Plaque support en aluminium (cellule de charge) usinée.

3.3.5. Assemblage et instrumentation

Le banc assemblé réunit la motorisation, la mesure de force et l'acquisition optique. La force est mesurée par une cellule de charge à jauges TedeA-Huntleigh 1004 (portée 300 g), lue par un convertisseur 24 bits HX711. L'étalonnage donne $1,5150 \times 10^{-3}$ mN par point brut (la force est positive en traction). Le moteur NEMA 23 est piloté par une carte powerSTEP01 (shield X-NUCLEO-IHM03A1) en micro-pas 1/32. Les capteurs de fin de course protègent les extrémités de la course. Le câblage de l'ensemble est donné ci-dessous.

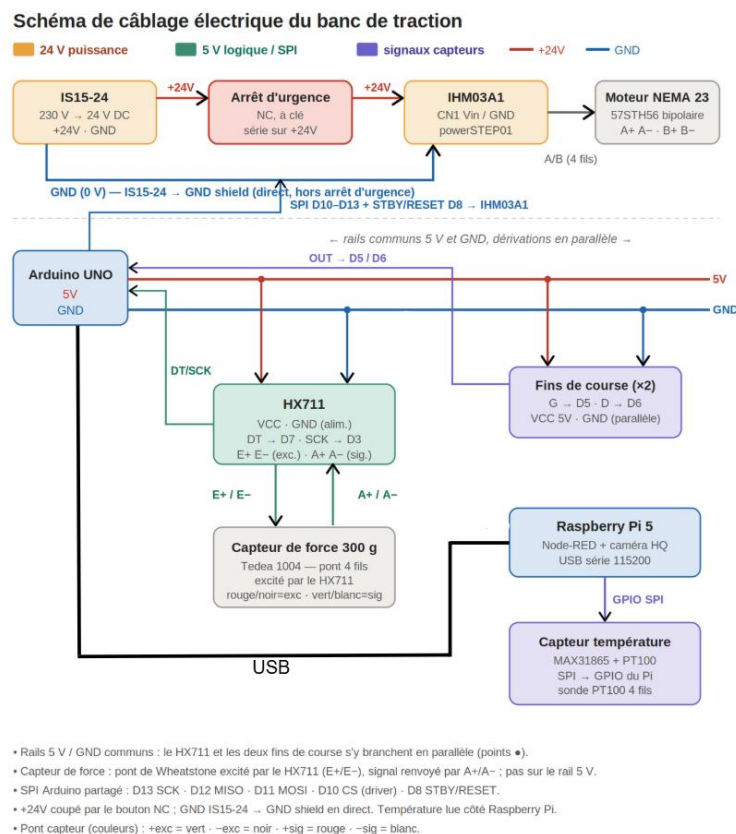


Figure 35 — Schéma de câblage du banc (moteur, carte de pilotage, cellule de charge et capteurs).

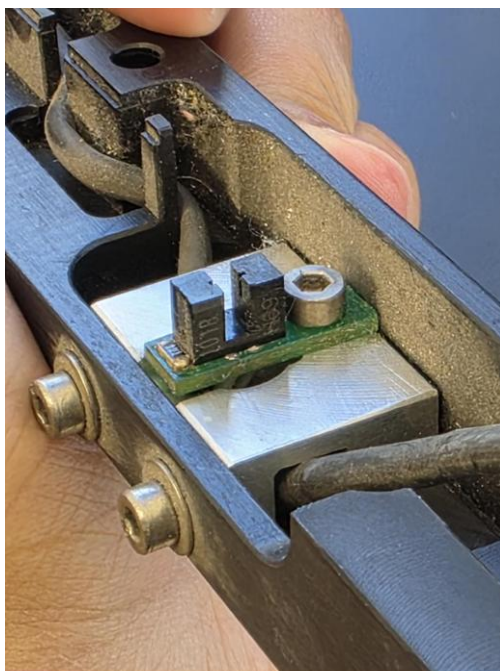


Figure 36 — Capteur de fin de course.

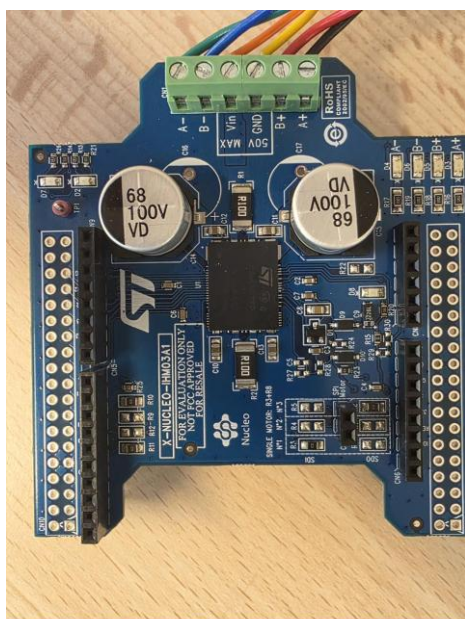


Figure 37 — Carte de pilotage moteur (IHM03A1 / powerSTEP01).



Figure 38 — Alimentation de puissance 24 V.

L'acquisition et le pilotage sont assurés par un Raspberry Pi 5 et une caméra HQ, avec le logiciel Node-RED. À chaque palier, une image est enregistrée et les données du capteur sont écrites dans un fichier CSV. Une sonde PT100 (via un module MAX31865) mesure la température du bain.

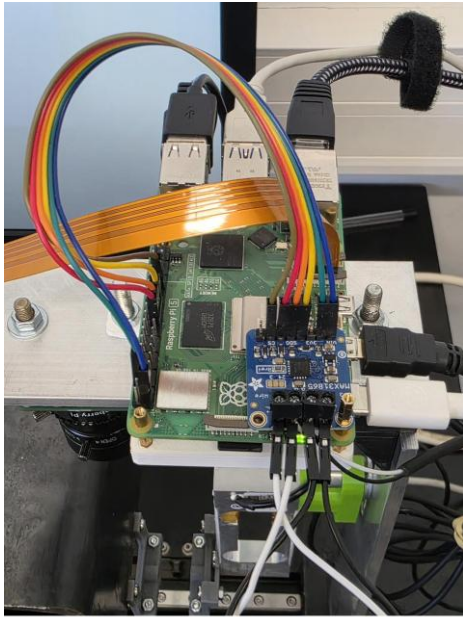


Figure 39 — Raspberry Pi 5 (acquisition et pilotage).

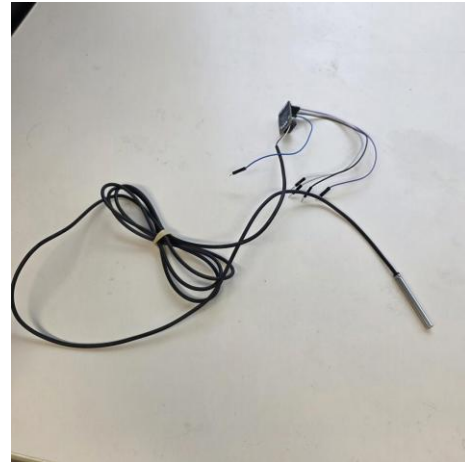


Figure 40 — Sonde de température PT100.

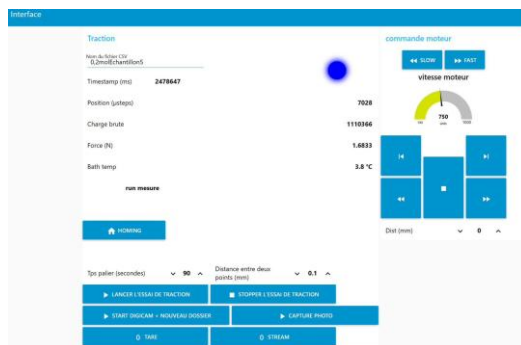


Figure 41 — Interface de pilotage Node-RED.

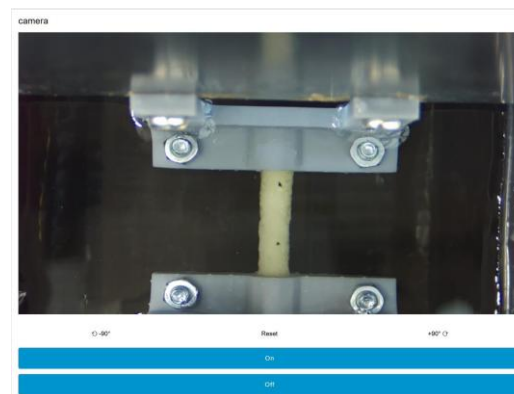


Figure 42 — Interface de prévisualisation de la caméra.

Enfin, j'ai ajouté un arrêt d'urgence matériel (bouton à verrouillage) sur la ligne d'alimentation 24 V du moteur. Cette protection est indispensable : l'arrêt logiciel est trop lent pour éviter une collision, et un incident de surcourse avait déjà endommagé une cellule de charge. L'arrêt d'urgence coupe tout de suite la puissance ; c'est la seule protection vraiment efficace.

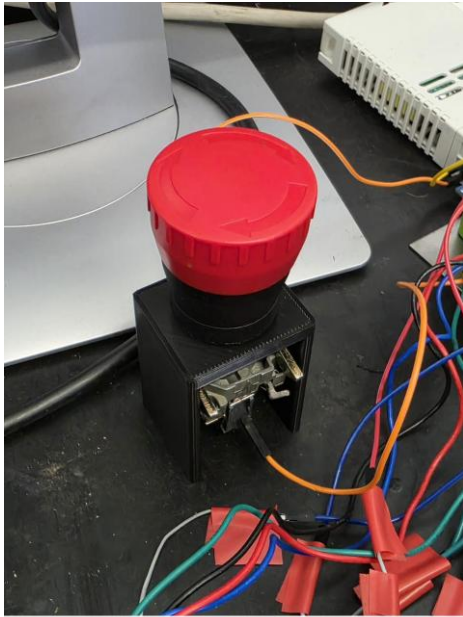


Figure 43 — Arrêt d'urgence matériel sur la ligne 24 V.

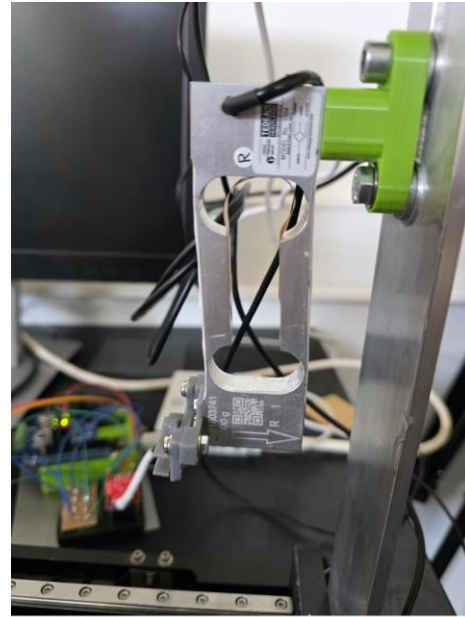


Figure 44 — Cellule de charge endommagée lors d'un incident de surcourse.

3.3.6. Étalonnage de la cellule de charge

Avant la campagne, la cellule de charge a été étalonnée pour relier la valeur brute du convertisseur HX711 à une force réelle. J'ai utilisé un étalonnage à un point, à partir d'une masse de référence connue. La cellule est d'abord tarée (mise à zéro). On applique ensuite la masse de référence, mesurée à la balance de précision (155,776 g), et on relève la valeur brute correspondante. Le rapport entre la masse et la valeur brute donne directement le facteur d'étalonnage, en supposant la réponse de la cellule linéaire et passant par zéro.

On obtient un facteur de 0,00015443 g par point brut. En multipliant par $g = 9,81 \text{ m/s}^2$, on convertit en force : un point brut correspond à $1,5150 \times 10^{-3} \text{ mN}$. La valeur brute augmente quand on tire sur la cellule : la force est donc positive en traction, sans signe négatif à appliquer. Ce facteur est ensuite utilisé dans tout le traitement des données. Un étalonnage à plusieurs masses constituerait une amélioration possible pour vérifier la linéarité sur toute la plage.

3.3.7. Réglages et validation du banc

Plusieurs réglages ont été nécessaires pour rendre le banc fiable et reproductible. La conversion entre les pas du moteur et le déplacement réel a été mesurée sur le banc : l'échelle est de 2510 micro-pas par millimètre, en micro-pas 1/32. Ce micro-pas a été choisi pour obtenir un mouvement régulier et silencieux.

La prise d'origine (homing) se fait en trois temps : le chariot va d'abord jusqu'à la butée droite, puis il dégage légèrement la butée, et enfin il effectue un déplacement de 148 mm vers la gauche pour se placer au point de départ de l'essai. Un arrêt automatique stoppe l'essai après 6,0 mm de déplacement (soit 15 060 micro-pas), ce qui évite d'endommager le montage.

Un réglage important concerne l'enregistrement des données. Au départ, les données n'étaient envoyées que pendant le mouvement du moteur ; or ces points sont parasités par les transitoires

électriques du moteur, et la force stabilisée du palier (celle qui sert au calcul de E) n'était pas enregistrée. J'ai corrigé ce point : le firmware continue d'envoyer des données pendant tout le palier, moteur à l'arrêt, ce qui donne un signal propre. De plus, l'horloge de l'essai est continue, ce qui permet de calculer correctement la vitesse côté MATLAB. Enfin, l'arrêt d'urgence matériel décrit plus haut complète la sécurité du banc.

3.3.8. Préparation des échantillons et protocole expérimental

Les échantillons sont des cylindres de pomme de terre droits ($\varnothing \approx 5$ mm). Je les ai découpés à l'aide d'une paille métallique dans des pommes de terre fraîches issues du même lot, puis je les ai coupés à 30 mm. La longueur effective L_0 utilisée pour le calcul correspond à la longueur visible par la caméra entre les mâchoires (≈ 16 – 17 mm), car la partie maintenue par les mâchoires ne se déforme pas. Le protocole est le suivant :

- Préparer les solutions de sorbitol pour chaque concentration. La masse est calculée par $m = C \times V \times M$ ($V = 0,400$ L ; $M = 182,17$ g/mol), pesée à la balance de précision puis dissoute.
- Tremper les échantillons environ 7 h (durée de trempage contrôlée et notée pour chaque échantillon). Les échantillons sont séparés dans des compartiments à parois en plexiglas à l'intérieur du contenant.
- Préparer les bains de glace par moulage, puis placer l'échantillon dans les mors. L'ensemble est immergé pour l'essai à froid (2 – 8 °C, toujours < 10 °C, suivi par la sonde PT100).
- Lancer l'essai sous Node-RED : tarage du capteur, paliers de $0,1$ mm, attente de 90 s, arrêt automatique à $6,0$ mm, avec capture d'images et enregistrement CSV.

Le sorbitol remplace le mannitol utilisé l'année précédente. Ces deux sucres-alcools ont la même masse molaire et ne pénètrent pas dans les cellules : ils sont donc presque équivalents pour l'osmose. Utiliser un seul soluté sur toute la campagne rend les comparaisons plus sûres.



Figure 45 — Pesée du sorbitol à la balance de précision.

Figure 46 — Contenant à compartiments (parois plexiglas) pour le trempage.



Figure 47 — Moule à glace utilisé pour l'essai à froid.



Figure 48 — Échantillon en cours d'essai.

3.3.9. Acquisition et traitement des données

Cette partie décrit comment les données sont mesurées, obtenues puis évaluées. Chaque essai produit deux jeux de données : un fichier CSV (horodatage, position du moteur, force, valeur brute de charge), écrit par Node-RED sur le Raspberry Pi, et une série d'images horodatées, une par palier. Le traitement se fait ensuite sous MATLAB (Image Processing Toolbox).

Détection des paliers. Le programme repère les paliers à partir de la vitesse de la position du moteur : tant que la position avance, on est dans un mouvement ; quand elle reste constante, on est dans un palier. Pour chaque palier, on ignore la première moitié des points (le mouvement, bruité par le moteur) et on moyenne la seconde moitié (le palier stabilisé, propre). On obtient ainsi, pour chaque palier, une force moyenne et un déplacement moyen.

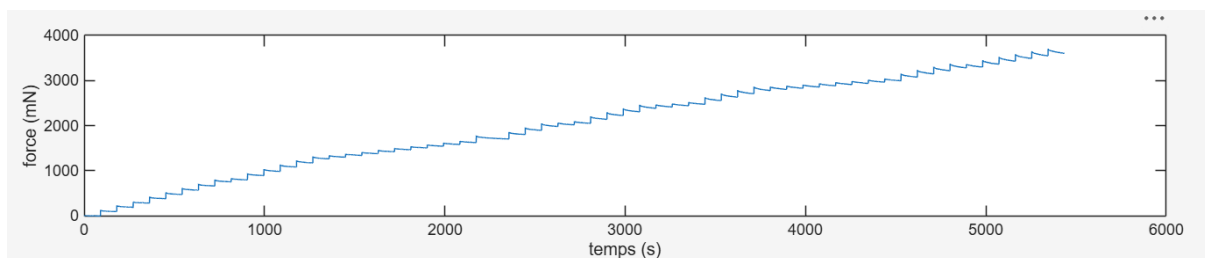


Figure 49 — Allure en escalier de la force au cours de l'essai : chaque palier correspond à un état stationnaire (moteur à l'arrêt) ; la seconde moitié de chaque palier est moyennée.

Calcul du module. On calcule d'abord la déformation ε , la contrainte σ et la section A :

$$\varepsilon = \frac{\Delta L}{L_0}$$

$$\sigma = \frac{F}{A}$$

$$A = \frac{\pi d^2}{4}$$

Le module de Young est le rapport de la contrainte sur la déformation. En pratique, on prend la pente de la courbe force-déplacement, ce qui donne :

$$E = \frac{\sigma}{\varepsilon} = \frac{\text{pente} \times L_0}{A}$$

Ici, $A = \pi(d/2)^2$ avec d le diamètre gonflé après trempage, et L_0 la longueur utile. La pente est calculée sur une zone fixe (les 30 premiers paliers) pour tous les échantillons. La valeur de L_0 est très importante : c'est la longueur visible entre les mors ($\approx 16-17$ mm) et non l'écart entre les mors (≈ 31 mm). Si on prenait l'écart entre les mors, E serait surestimé d'un facteur $\approx 1,85$. La figure ci-dessous montre L_0 et la distance entre les deux points marqués qui sert à mesurer l'allongement réel.

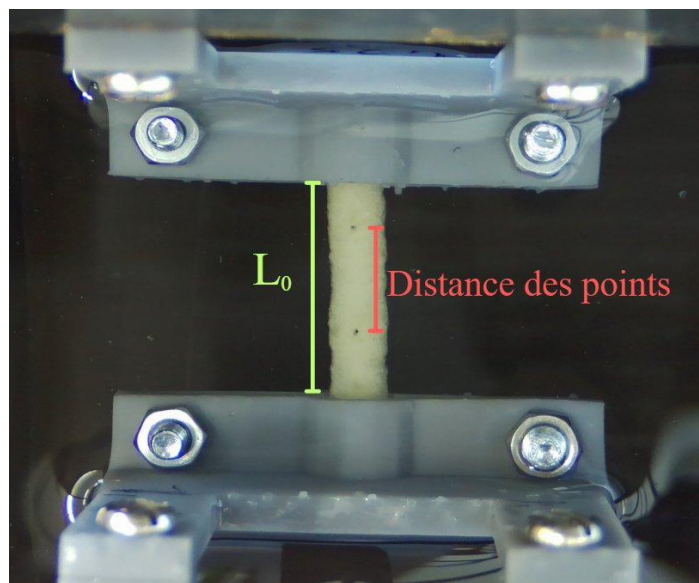


Figure 50 — Échantillon entre les mors vis-écrou : longueur utile L_0 et distance entre les deux points suivis pour l'allongement réel.

Module apparent et module corrigé. On distingue deux mesures du module. Le module apparent vient du déplacement du moteur ; c'est le livrable principal. Le module corrigé vient de l'allongement réel mesuré sur les images, qui ne dépend pas du glissement aux mors. La part de course perdue (59 à 93 % selon les essais) vient à la fois du glissement aux mors et de la flexion de la cellule de charge : une partie du déplacement du moteur n'arrive donc pas à l'échantillon. Le module corrigé reste valable, car les points marqués mesurent directement la vraie déformation du matériau ; on le donne en ordre de grandeur.

Mesure de l'allongement réel. Pour le module corrigé, j'utilise un pointage manuel : sur six images réparties sur les paliers, je clique les deux points marqués sur l'échantillon (avec un zoom pour viser plus précisément). La distance entre ces deux points donne l'allongement réel. Le suivi automatique des points a été abandonné, car il s'accrochait parfois aux mors au lieu des vrais points ; la comparaison avec le pointage manuel a montré ce défaut. Le diamètre mesuré sur l'image a aussi été comparé au pied à coulisse (écart $\approx +1,8$ %), ce qui confirme la valeur utilisée pour la section A .

Évaluation et sélection des essais. Tous les essais ne sont pas conservés. Un essai est écarté quand l'échantillon rompt pendant l'essai, quand il n'a pas été assez trempé (il paraît alors trop rigide), ou quand l'image est de trop mauvaise qualité pour le pointage. Par exemple, l'échantillon 1 à 0,1 mol/L (rupture) et l'échantillon 1 à 0,4 mol/L (sous-trempé puis rompu) ont été exclus. Pour chaque concentration, on calcule ensuite la moyenne et l'écart-type du module sur les essais retenus. Ce sont ces valeurs qui forment les courbes du module en fonction de la concentration.

3.3.10. Résultats

La figure suivante montre le résultat principal : le module de Young en fonction de la concentration en sorbitol, avec le module apparent (déplacement moteur) et le module corrigé (pointage manuel).

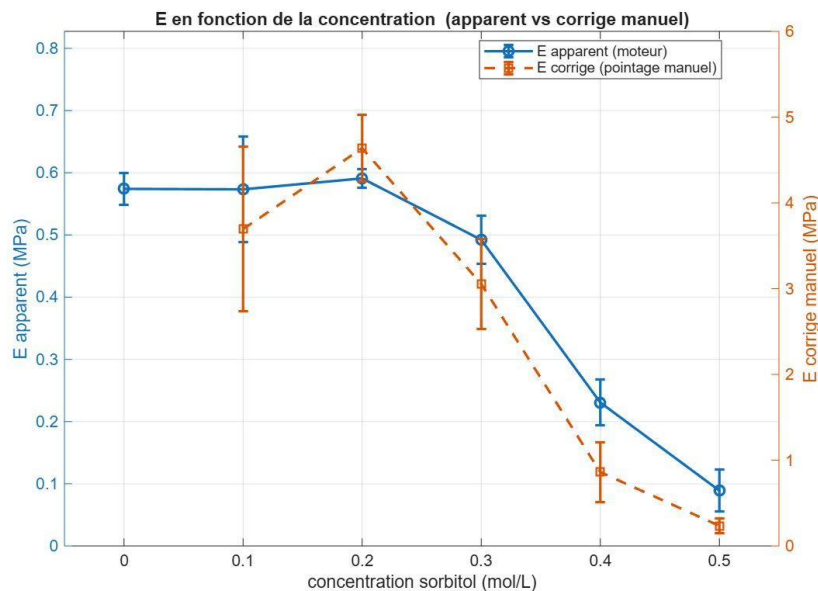


Figure 51 — Module de Young en fonction de la concentration en sorbitol : module apparent (axe de gauche) et module corrigé (axe de droite).

Le module apparent reste presque constant entre 0 et 0,2 mol/L (autour de 0,57–0,61 MPa), puis il baisse jusqu'à environ 0,09 MPa à 0,5 mol/L : le tissu devient environ 6 fois plus mou. Le module corrigé suit la même baisse, à un niveau de quelques MPa, proche de la littérature. Le tableau suivant donne les moyennes par concentration.

Concentration (mol/L)	n (apparent)	E apparent moyen (MPa)	E apparent écart-type	n (corrigé)	E corrigé moyen (MPa)	E corrigé écart-type
0.0	4	0.574	0.025	0		
0.1	5	0.606	0.026	5	3.960	0.785
0.2	3	0.591	0.016	3	4.638	0.389
0.3	5	0.492	0.039	5	3.055	0.524
0.4	5	0.231	0.037	5	0.725	0.102
0.5	5	0.089	0.033	5	0.235	0.089

Figure 52 — Moyennes du module de Young par concentration (apparent et corrigé).

Le détail de chaque échantillon est donné ci-dessous (et repris en annexe).

Échantillon	Concentration (mol/L)	L0 (mm)	d (mm)	A (mm²)	E apparent (MPa)	E corrigé manuel (MPa)	Perte de course (%)	Retenu (1/0)	Remarque
0molEchantillon1	0.0	16.70	4.920	19.01	0.547			1	
0molEchantillon2	0.0	16.66	4.970	19.40	0.590			1	Trempage ~3 h (sous-trempe) — à confirmer
0molEchantillon3	0.0	16.50	5.010	19.71	0.559			1	Optique médiocre (sans effet sur E apparent)
0molEchantillon5	0.0	16.46	5.010	19.71	0.601			1	
0,1molEchantillon1	0.1	16.30	5.020	19.79	0.408	2.365	87	0	Rupture en cours d'essai — exclu de la moyenne
0,1molEchantillon2	0.1	16.36	5.070	20.19	0.585	4.555	92	1	
0,1molEchantillon3	0.1	16.36	5.070	20.19	0.624	4.100	93	1	
0,1molEchantillon4	0.1	16.66	4.994	19.59	0.584	2.623	82	1	
0,1molEchantillon5	0.1	16.72	5.047	20.01	0.595	4.010	87	1	
0,1molEchantillon6	0.1	16.74	5.025	19.83	0.643	4.511	89	1	
0,2molEchantillon1	0.2	17.33	4.760	17.80	0.586	4.297	89	1	
0,2molEchantillon2	0.2	16.99	4.875	18.67	0.608	5.061	88	1	
0,2molEchantillon3	0.2	16.65	4.770	17.87	0.578	4.555	89	1	
0,3molEchantillon1	0.3	17.18	4.673	17.15	0.477	3.931	93	1	
0,3molEchantillon2	0.3	17.03	4.650	16.98	0.432	2.997	91	1	
0,3molEchantillon3	0.3	17.02	4.660	17.06	0.532	2.754	84	1	
0,3molEchantillon4	0.3	17.02	4.612	16.71	0.511	2.570	81	1	
0,3molEchantillon5	0.3	16.83	4.630	16.84	0.508	3.022	83	1	
0,4molEchantillon1	0.4	16.87	4.400	15.21	0.280	1.542	81	0	Trempage ~5-6 h (sous-trempe) + rupture — exclu
0,4molEchantillon2	0.4	16.93	4.270	14.32	0.239	0.847	75	1	
0,4molEchantillon3	0.4	16.72	4.390	15.14	0.198	0.678	74	1	
0,4molEchantillon4	0.4	16.60	4.440	15.48	0.190	0.588	73	1	
0,4molEchantillon5	0.4	16.62	4.500	15.90	0.248	0.709	68	1	
0,4molEchantillon6	0.4	16.67	4.520	16.05	0.279	0.801	68	1	
0,5molEchantillon1	0.5	16.91	4.420	15.34	0.058	0.159	63	1	
0,5molEchantillon2	0.5	16.78	4.450	15.55	0.093	0.226	59	1	
0,5molEchantillon3	0.5	16.85	4.325	14.69	0.058	0.162	64	1	
0,5molEchantillon4	0.5	16.87	4.355	14.90	0.096	0.249	63	1	
0,5molEchantillon5	0.5	16.94	4.330	14.73	0.139	0.377	66	1	

Figure 53 — Données par échantillon (tous les essais).

Les courbes contrainte-déformation confirment ce comportement : la pente du début, qui donne le module, baisse nettement aux fortes concentrations. La figure suivante regroupe tous les essais retenus, puis les courbes par concentration sont présentées une à une.

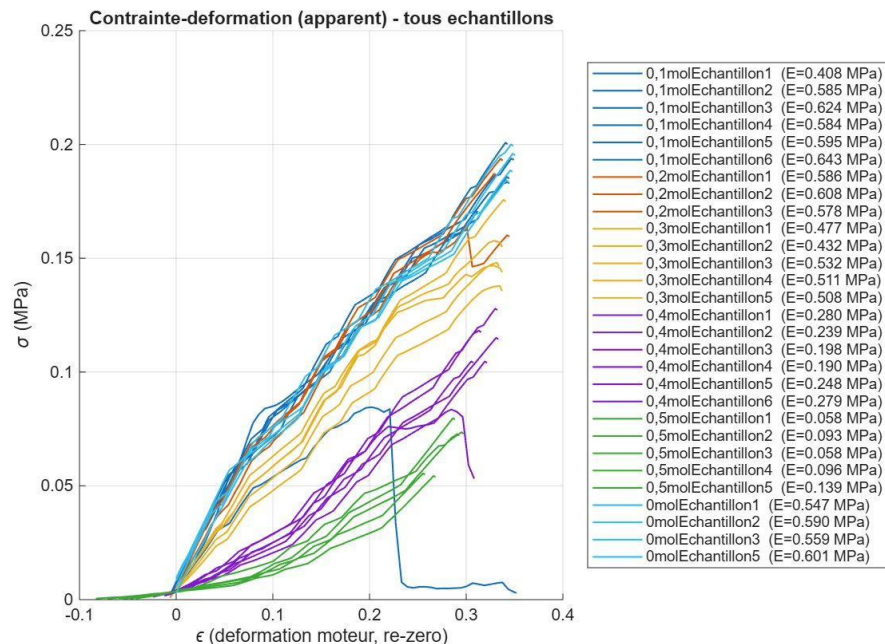


Figure 54 — Courbes contrainte-déformation de tous les essais retenus.

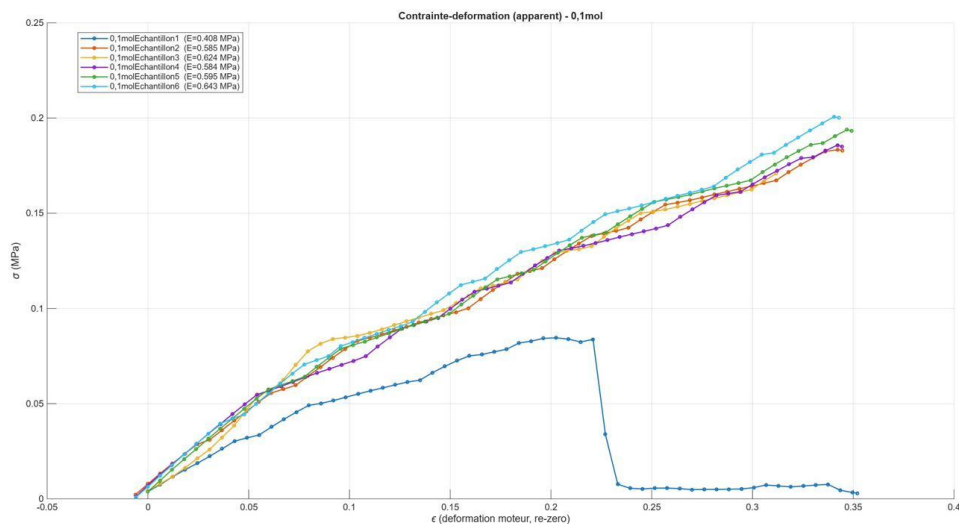


Figure 55 — Courbes contrainte-déformation à 0,1 mol/L.

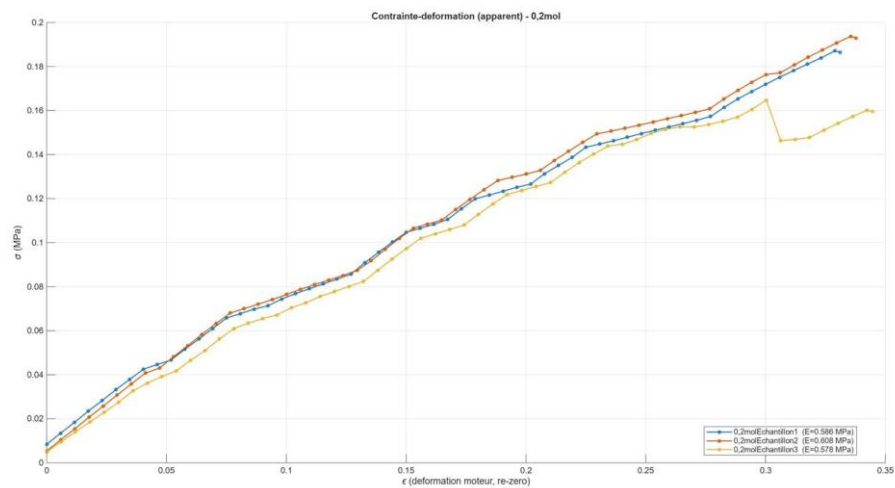


Figure 56 — Courbes contrainte-déformation à 0,2 mol/L.

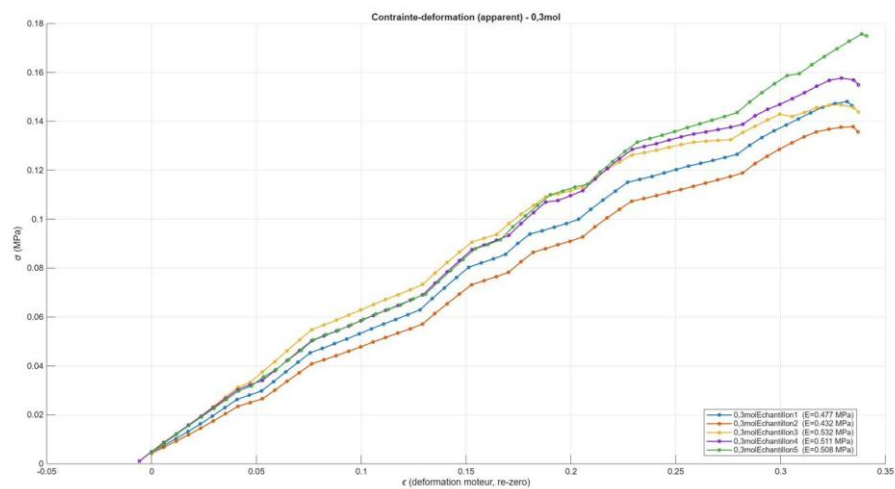


Figure 57 — Courbes contrainte-déformation à 0,3 mol/L.

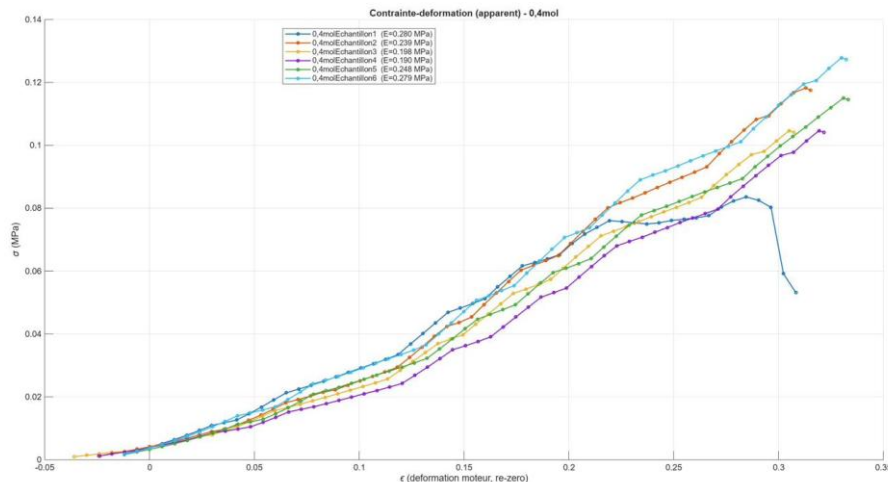


Figure 58 — Courbes contrainte-déformation à 0,4 mol/L.

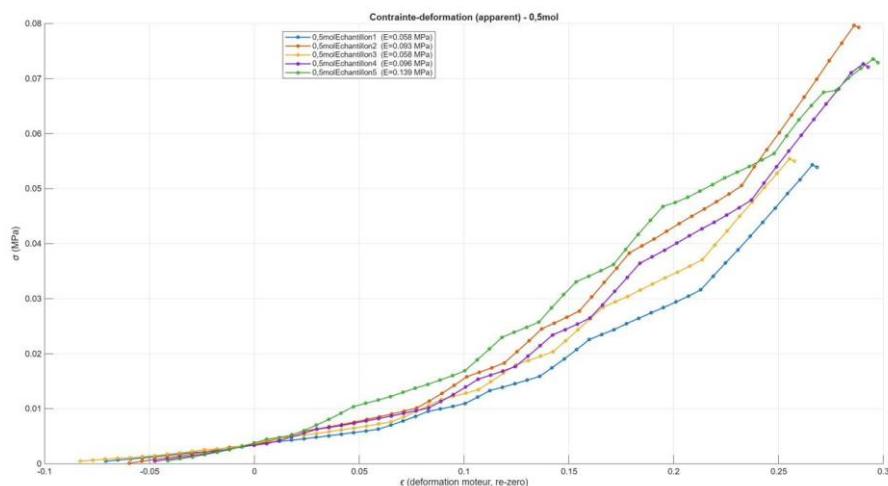


Figure 59 — Courbes contrainte-déformation à 0,5 mol/L.

Le coefficient de Poisson n'a pu être estimé que de façon approximative. Avec une seule caméra, la variation de diamètre est trop faible et proche du bruit. On obtient seulement des bornes basses (de l'ordre de 0,03 à 0,28). On présente donc cette mesure comme une limite du dispositif, pas comme un résultat précis.

3.3.11. Discussion

Le résultat principal — un module qui baisse quand la concentration augmente — est le comportement attendu pour un solide cellulaire pressurisé. À concentration nulle (eau pure), la turgescence est maximale, donc le tissu est le plus rigide. Quand la concentration augmente, l'eau sort des cellules, la turgescence baisse et le tissu devient plus mou. Le plateau de 0 à 0,2 mol/L correspond à un tissu encore turgescant. La baisse à partir de 0,3 mol/L montre que l'on s'approche du point isotonique, où la turgescence se vide peu à peu.

Cette tendance est cohérente avec la figure de référence du laboratoire (module qui augmente avec la turgescence) : lue dans l'autre sens, une rigidité qui augmente avec la turgescence revient à une rigidité qui baisse avec la concentration. Elle est aussi compatible avec les ordres

de grandeur de la littérature (de quelques dixièmes à quelques MPa). On note un écart avec le rapport du stagiaire précédent, qui trouvait une rigidité qui augmente avec la concentration. L'effet de durcissement par déshydratation n'apparaît que bien après la plasmolyse ; il ne doit donc pas se voir dans la plage étudiée, où c'est la turgescence qui commande.

Les principales sources d'erreur sont : la perte de course (glissement aux mors et flexion de la cellule), qui justifie le module corrigé ; les petites différences entre échantillons biologiques ; et la durée de trempage, qui est sensible (un échantillon pas assez trempé paraît trop rigide). Enfin, les essais sont faits à froid (2–8 °C), alors que la plupart des valeurs publiées sont à température ambiante. Les valeurs absolues ne sont donc comparables qu'à titre indicatif : c'est la tendance qui est le résultat le plus solide.

3.4. Bilan technique et perspectives

Sur le plan technique, j'ai conçu, fabriqué et instrumenté un nouveau banc d'essai (mesure de force, acquisition optique, pilotage Node-RED, sécurité), puis j'ai réalisé une campagne complète sur la série 0–0,5 mol/L de sorbitol. Le livrable principal — la courbe du module de Young apparent en fonction de la concentration — est clair et s'explique par la physique : plateau jusqu'à 0,2 mol/L, puis un tissu environ 6 fois plus mou. Le module corrigé confirme cette tendance.

Les perspectives sont les suivantes :

- Améliorer la mesure du coefficient de Poisson avec une deuxième caméra ou une vue rapprochée de la zone utile, car le signal latéral est aujourd'hui trop faible.
- Étendre la série aux concentrations 0,6–0,7 mol/L pour étudier le plateau de plasmolyse.
- Réduire encore la perte de course (mors anti-glissement, chaîne de mesure plus rigide) et fixer une durée de trempage standard.
- Construire un caisson fermé avec un éclairage suffisant et régulier autour de l'échantillon. Des images plus nettes et plus stables rendraient le système optique fiable et permettraient d'automatiser la mesure du coefficient de Poisson (suivi automatique du diamètre).
- Mesurer directement l'allongement de l'échantillon pendant qu'on le tire (suivi des points sur l'image), plutôt que de se fier au déplacement du moteur qui subit le glissement. Avec un système d'imagerie fiable, cette mesure de l'allongement réel pourrait être automatisée pour chaque essai.

4. Rapport PPP

4.1. Description d'un métier : doctorant en biomécanique (ISM)

Pour le Projet Personnel et Professionnel, j'ai mené un entretien le 22 avril 2026 avec M. Thomas GUILLANTON, doctorant employé par l'ISM à la halle d'essai. Sa thèse porte sur l'étude du mouvement du crabe, plus précisément la locomotion de *Carcinus aestuarii*.

Pour étudier la locomotion du crabe, il utilise un dispositif qui combine l'imagerie par thermographie et la corrélation d'images en 3D (3D DIC) : le crabe est suspendu dans l'air, tenu par de la pâte à modeler, pendant que les caméras analysent son mouvement. Il relie ensuite ces observations à l'anatomie de l'animal : les crabes sont sacrifiés par électrocution, puis envoyés à un laboratoire de biologie qui réalise l'analyse anatomique. Les crabes sont gardés vivants dans un bac dont il contrôle les conditions (teneur de l'eau en ammoniac et en nitrate, oxygénation par un bulleur, etc.).

Cet entretien montre un métier de la recherche à la frontière de la mécanique, de l'instrumentation optique et de la biologie. Il m'a aidé à mieux comprendre la démarche scientifique et la variété des compétences utilisées à l'ISM.

QOQCP	Synthèse de l'entretien
Quoi ?	Étudier la locomotion du crabe <i>Carcinus aestuarii</i> (thèse en biomécanique).
Qui ?	Doctorant de l'ISM, encadré par ses directeurs de thèse, en lien avec un laboratoire de biologie partenaire.
Où ?	Halle d'essai de l'ISM / IUT d'Aix-en-Provence.
Quand ?	Pendant sa thèse. Entretien réalisé le 22/04/2026.
Comment ?	Thermographie + corrélation d'images 3D (DIC) ; analyse anatomique après sacrifice ; animaux gardés en conditions contrôlées.
Pourquoi ?	Comprendre les mécanismes de la locomotion, dans une logique de recherche bio-inspirée.

4.2. Activités annexes

En plus de ma mission, j'ai aidé l'équipe de l'ISM à ranger et à désencombrer la halle d'essai. Je suis surtout intervenu au second niveau (plateforme sur échafaudage) pour évacuer les rebuts et ranger la visserie inutilisée dans ses compartiments. J'ai aussi aidé à ranger la salle InProto, en triant le matériel et en classant des moteurs inutilisés par compartiments.

5. Conclusion générale et personnelle

Ce stage a été une expérience riche et variée. Sur le plan technique, il m'a fait découvrir un domaine nouveau pour moi : l'électronique. J'ai appris à utiliser des microcontrôleurs (Arduino UNO R3, Raspberry Pi 5) et à assembler le système optique, le système de température, le système de fins de course et le système de mesure de force, puis à programmer et à piloter un moteur pas-à-pas NEMA 23. Comme j'avais rarement utilisé la programmation dans des projets concrets, j'ai vraiment progressé.

J'ai aussi mis en pratique et approfondi mes compétences en fabrication. En usinage, j'ai programmé une commande numérique (ShopMill, SOMAB 250+) pour réaliser l'extension de l'arbre du moteur, et j'ai usiné par tournage conventionnel le trou oblong du support de la cellule de charge pour permettre un réglage en hauteur. J'ai aussi découvert l'impression 3D en résine sur Form3L et ses étapes : nettoyer les pièces à l'alcool, puis les durcir aux UV pour atteindre leurs propriétés finales.

Plusieurs difficultés m'ont fait progresser. Pour rendre le mouvement du moteur régulier, le débogage a demandé de trouver la bonne configuration : grâce aux exemples de M. Louison FIORE, le moteur est devenu fluide en micro-pas 1/32. L'usinage a aussi été exigeant : j'avais oublié certaines procédures, et l'extension d'arbre demandait de bien maîtriser les tolérances pour obtenir un ajustement ajusté ; M. Eric JEANCELME m'a guidé pendant cette fabrication. Enfin, j'ai appris la soudure, que je n'avais jamais pratiquée, en réalisant un circuit parallèle sur une plaque d'essai (plutôt que sur une breadboard) pour éviter que les fils Dupont se débranchent.

Ce stage correspond bien aux compétences de mon parcours Conception de Produits Durables : analyser un besoin, concevoir en CAO, choisir les matériaux et les procédés, fabriquer et assembler un dispositif, dans une logique de sobriété (réemploi d'un guidage existant, impression 3D, moyens simples). Il confirme aussi mon projet de poursuite d'études en mécanique, où la conception, la fabrication et l'électronique vues ici me seront très utiles.

Bibliographie et webographie

- [1] CHAIX R. (2025). Caractérisation mécanique de solide cellulaire pressurisé : parenchyme de pomme de terre. Rapport de stage BUT GMP, ISM — IUT Aix-en-Provence.
- [2] LIN T.-T., PITT R. E. (1986). Rheology of apple and potato tissue as affected by cell turgor pressure. *Journal of Texture Studies*, 17, 291–313.
- [3] TADRIST L. et al. Document de présentation du sujet « Mechanical Characterization of Pressurized Cellular Solid: Potato Parenchyma », ISM — équipe SBI (2025–2026).
- [4] Jiali Fan. Essais sur tissu de pomme de terre par corrélation d'images (DIC). MDPI, Agriculture, 2022, 12(7), 1026. <https://www.mdpi.com/2077-0472/12/7/1026>
- [5] Institut des Sciences du Mouvement — équipe SBI. <https://ism.univ-amu.fr/>
- [6] NIKLAS K. J. (1992). *Plant Biomechanics: An Engineering Approach to Plant Form and Function*. University of Chicago Press (manuel de biomécanique végétale).
- [7] GIBSON L. J., ASHBY M. F. (1997). *Cellular Solids: Structure and Properties*, 2^e éd. Cambridge University Press.

Annexes

Annexe A — Données par échantillon

L_0 : longueur utile ; d : diamètre gonflé ; A : section ; perte : part de course perdue ; Ret. : 1 = retenu, 0 = exclu.

Échantillon	Concentration (mol/L)	L_0 (mm)	d (mm)	A (mm ²)	E apparent (MPa)	E corrigé manuel (MPa)	Perte de course (%)	Retenu (1/0)	Remarque
0molEchantillon1	0.0	16.70	4.920	19.01	0.547			1	
0molEchantillon2	0.0	16.66	4.970	19.40	0.590			1	Trempage ~3 h (sous-trempé) — à confirmer
0molEchantillon3	0.0	16.50	5.010	19.71	0.559			1	Optique médiocre (sans effet sur E apparent)
0molEchantillon5	0.0	16.46	5.010	19.71	0.601			1	
0,1molEchantillon1	0.1	16.30	5.020	19.79	0.408	2.365	87	0	Rupture en cours d'essai — exclu de la moyenne
0,1molEchantillon2	0.1	16.36	5.070	20.19	0.585	4.555	92	1	
0,1molEchantillon3	0.1	16.36	5.070	20.19	0.624	4.100	93	1	
0,1molEchantillon4	0.1	16.66	4.994	19.59	0.584	2.623	82	1	
0,1molEchantillon5	0.1	16.72	5.047	20.01	0.595	4.010	87	1	
0,1molEchantillon6	0.1	16.74	5.025	19.83	0.643	4.511	89	1	
0,2molEchantillon1	0.2	17.33	4.760	17.80	0.586	4.297	89	1	
0,2molEchantillon2	0.2	16.99	4.875	18.67	0.608	5.061	88	1	
0,2molEchantillon3	0.2	16.65	4.770	17.87	0.578	4.555	89	1	
0,3molEchantillon1	0.3	17.18	4.673	17.15	0.477	3.931	93	1	
0,3molEchantillon2	0.3	17.03	4.650	16.98	0.432	2.997	91	1	
0,3molEchantillon3	0.3	17.02	4.660	17.06	0.532	2.754	84	1	
0,3molEchantillon4	0.3	17.02	4.612	16.71	0.511	2.570	81	1	
0,3molEchantillon5	0.3	16.83	4.630	16.84	0.508	3.022	83	1	
0,4molEchantillon1	0.4	16.87	4.400	15.21	0.280	1.542	81	0	Trempage ~5-6 h (sous-trempé) + rupture — exclu
0,4molEchantillon2	0.4	16.93	4.270	14.32	0.239	0.847	75	1	
0,4molEchantillon3	0.4	16.72	4.390	15.14	0.198	0.678	74	1	
0,4molEchantillon4	0.4	16.60	4.440	15.48	0.190	0.588	73	1	
0,4molEchantillon5	0.4	16.62	4.500	15.90	0.248	0.709	68	1	
0,4molEchantillon6	0.4	16.67	4.520	16.05	0.279	0.801	68	1	
0,5molEchantillon1	0.5	16.91	4.420	15.34	0.058	0.159	63	1	
0,5molEchantillon2	0.5	16.78	4.450	15.55	0.093	0.226	59	1	
0,5molEchantillon3	0.5	16.85	4.325	14.69	0.058	0.162	64	1	
0,5molEchantillon4	0.5	16.87	4.355	14.90	0.096	0.249	63	1	
0,5molEchantillon5	0.5	16.94	4.330	14.73	0.139	0.377	66	1	

Annexe B — Dessins de définition (CATIA V5)

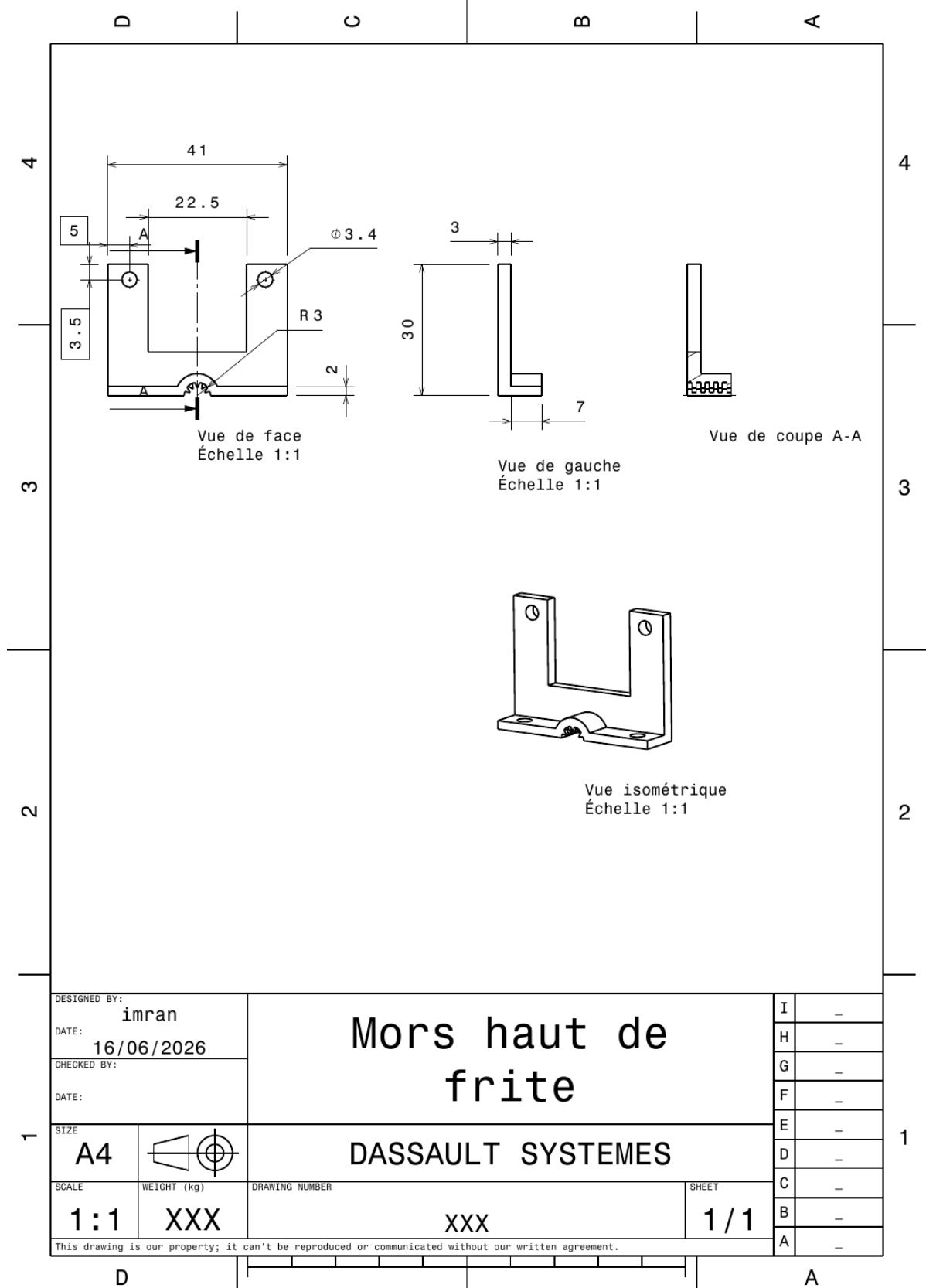


Figure 60 — Dessin de définition — Mors (partie haute).

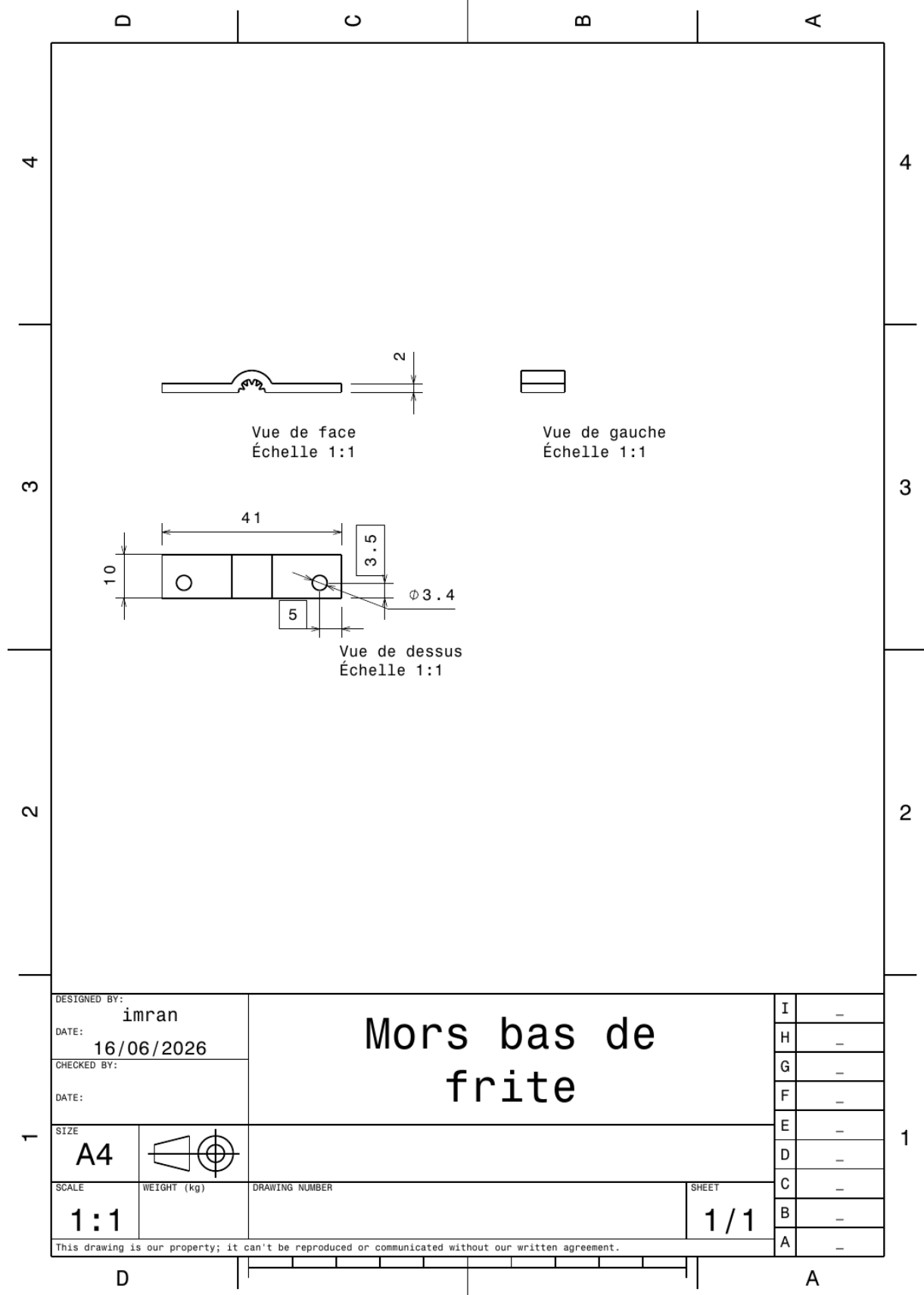


Figure 61 — Dessin de définition — Mors (partie basse).

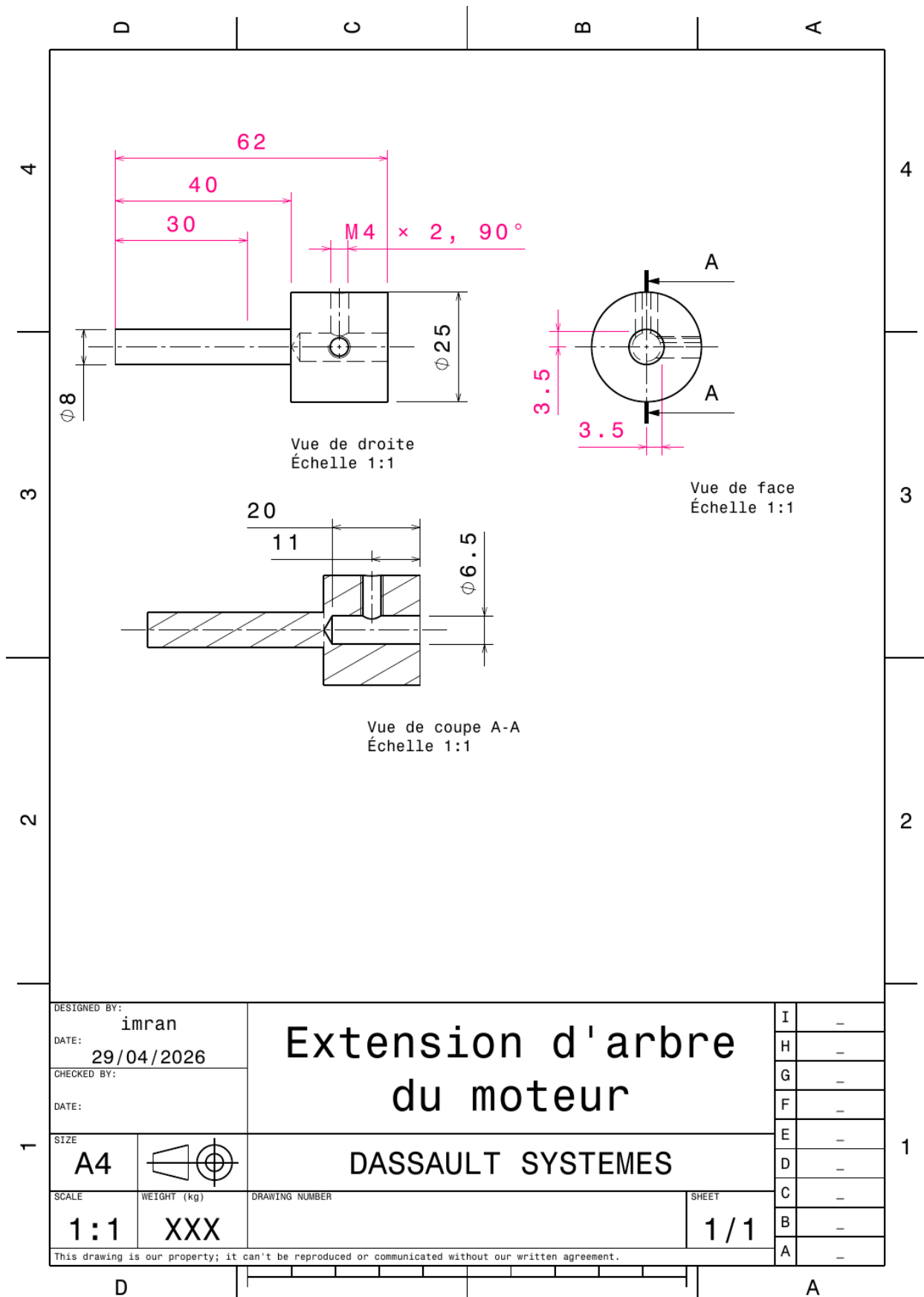


Figure 62 — Dessin de définition — Extension d'arbre du moteur.

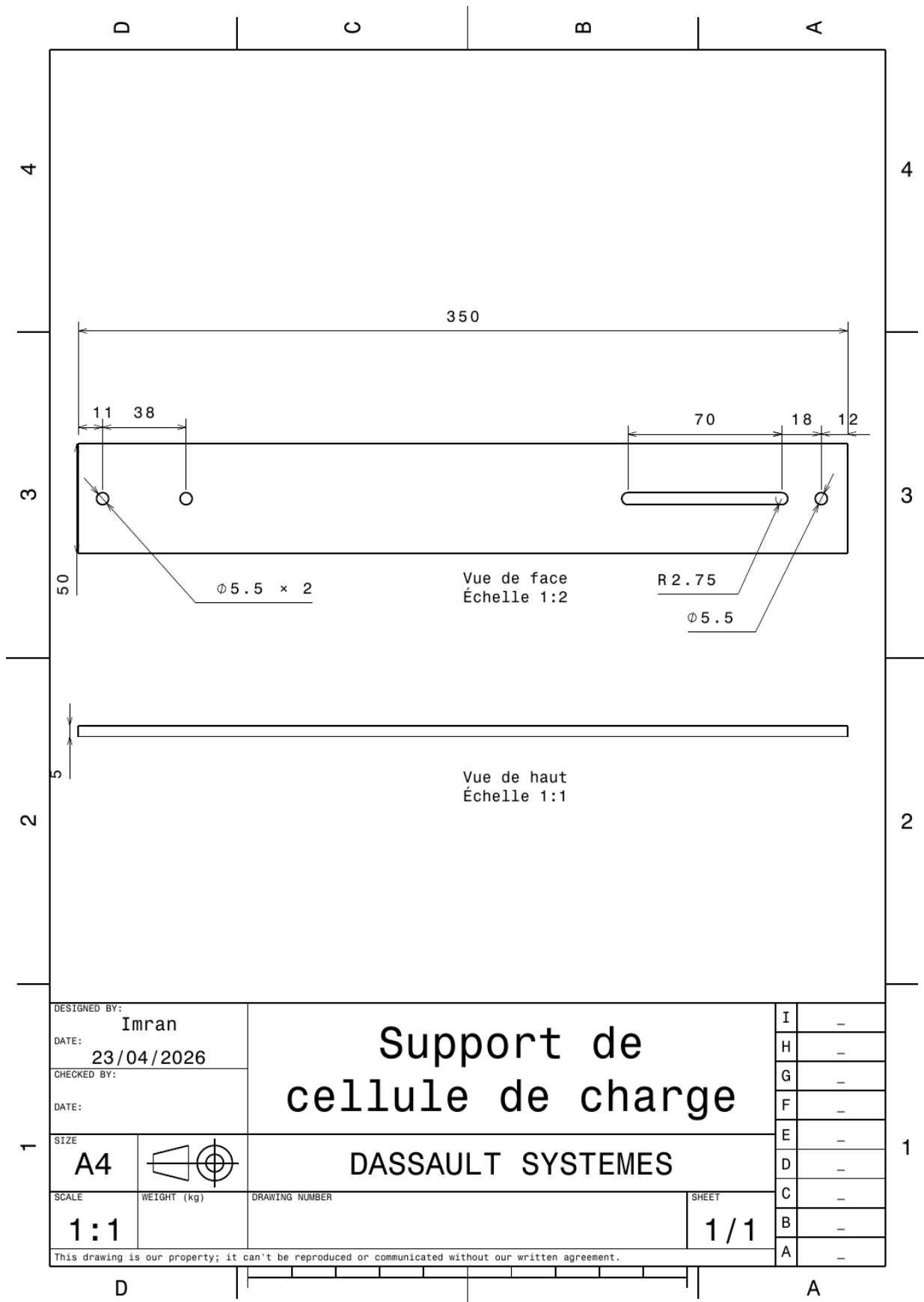


Figure 63 — Dessin de définition — Support de la cellule de charge.

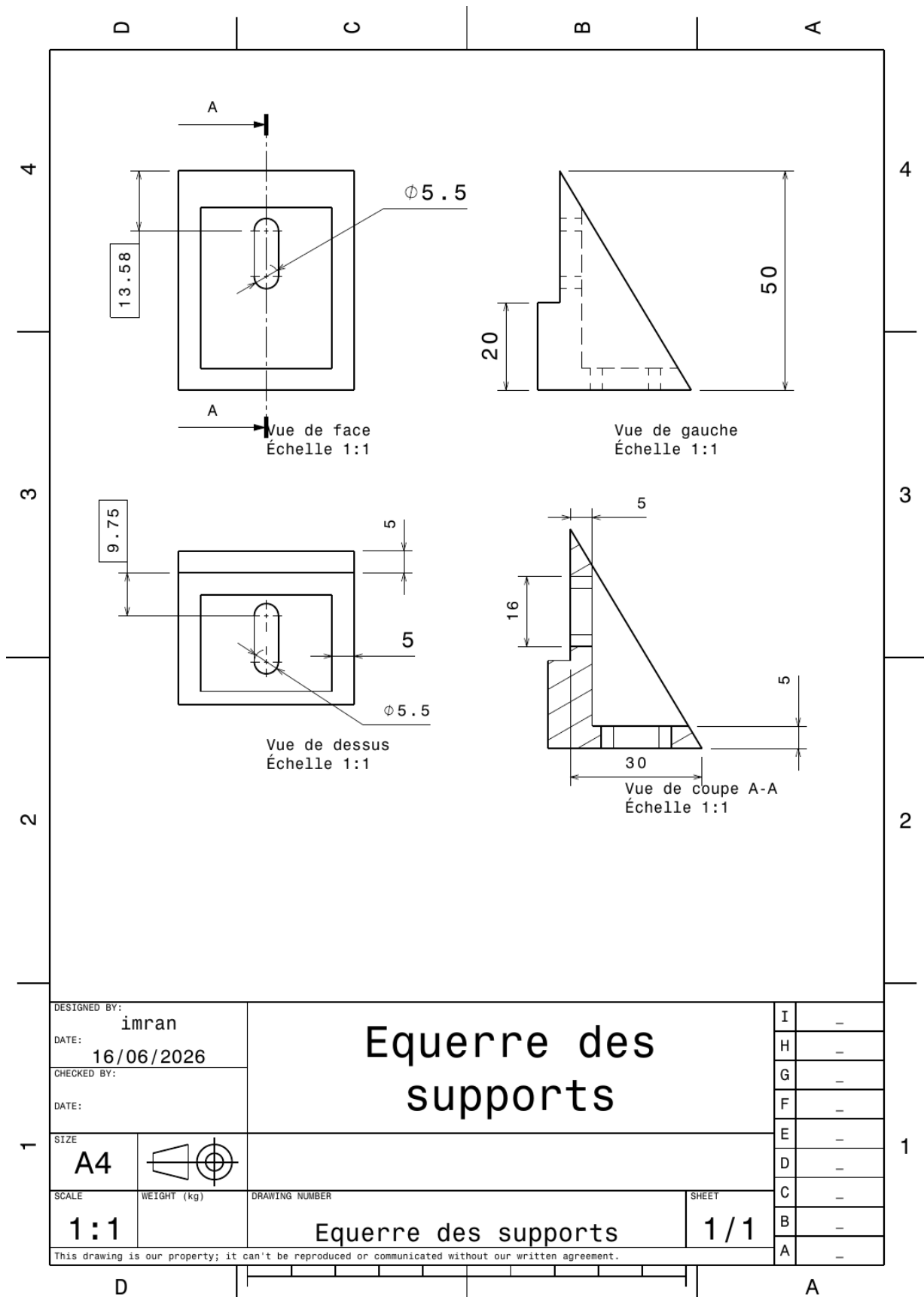


Figure 64 — Dessin de définition — Équerre des supports.

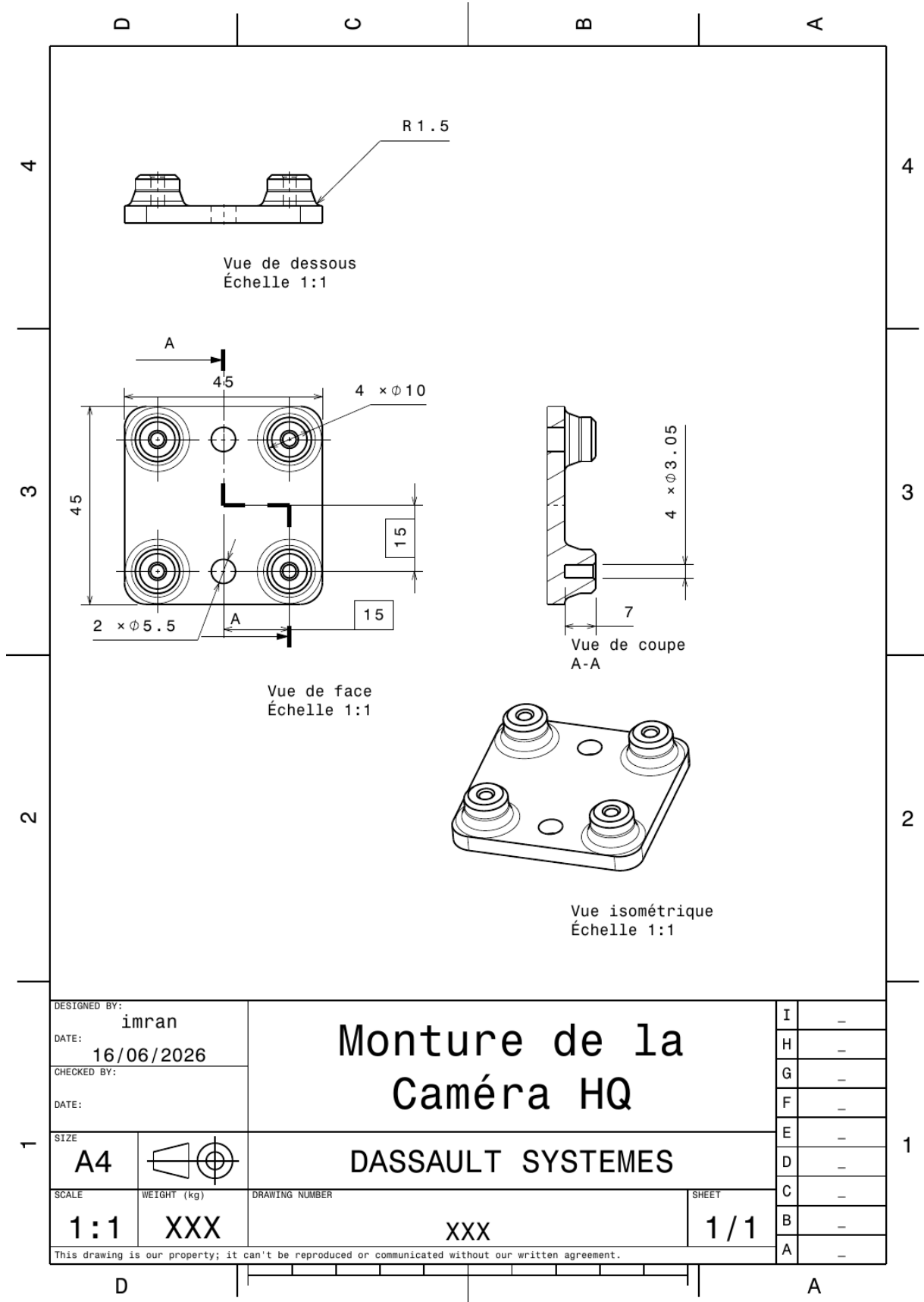


Figure 65 — Dessin de définition — Monture de la caméra.

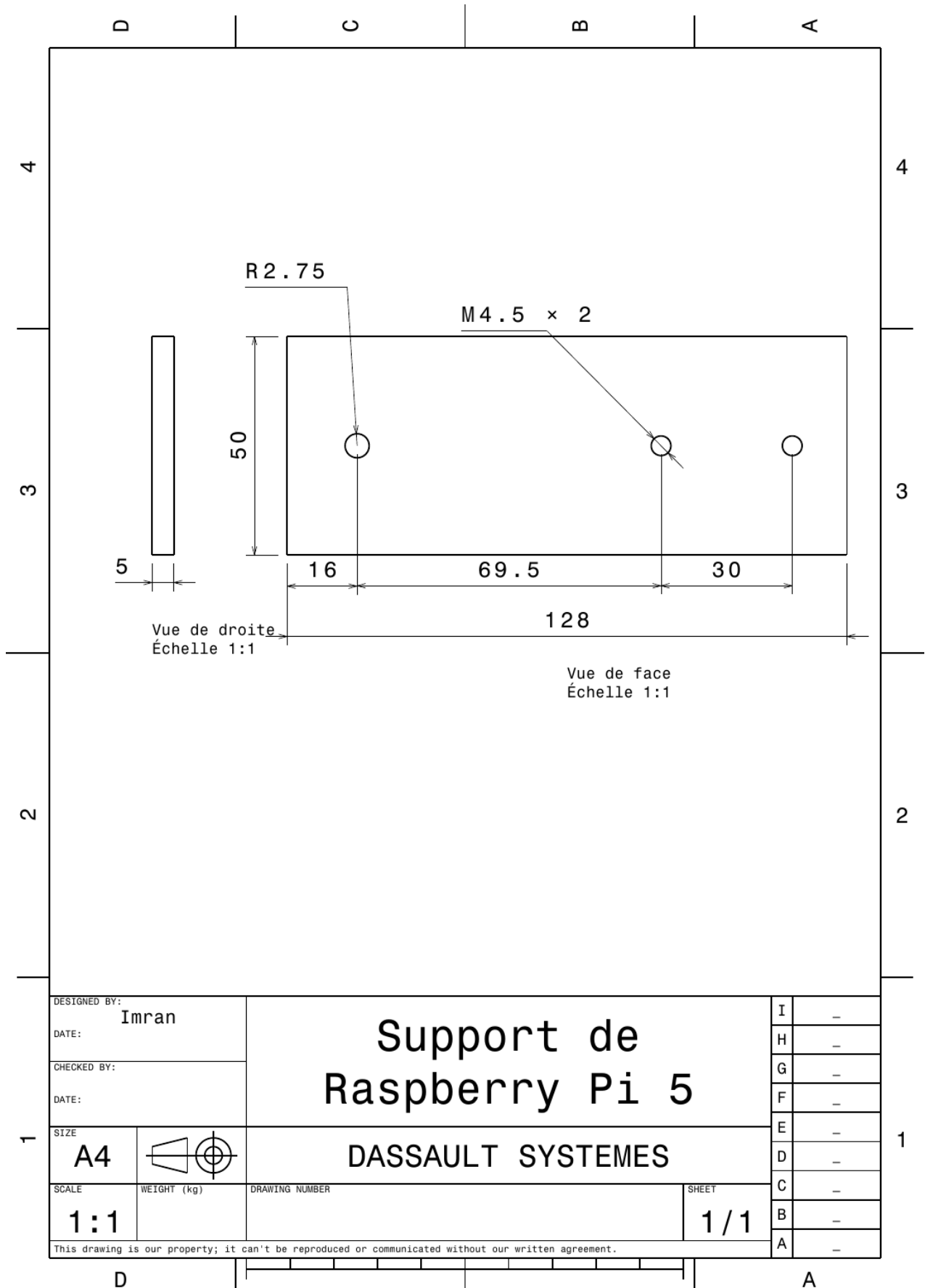


Figure 66 — Dessin de définition — Support du Raspberry Pi 5.

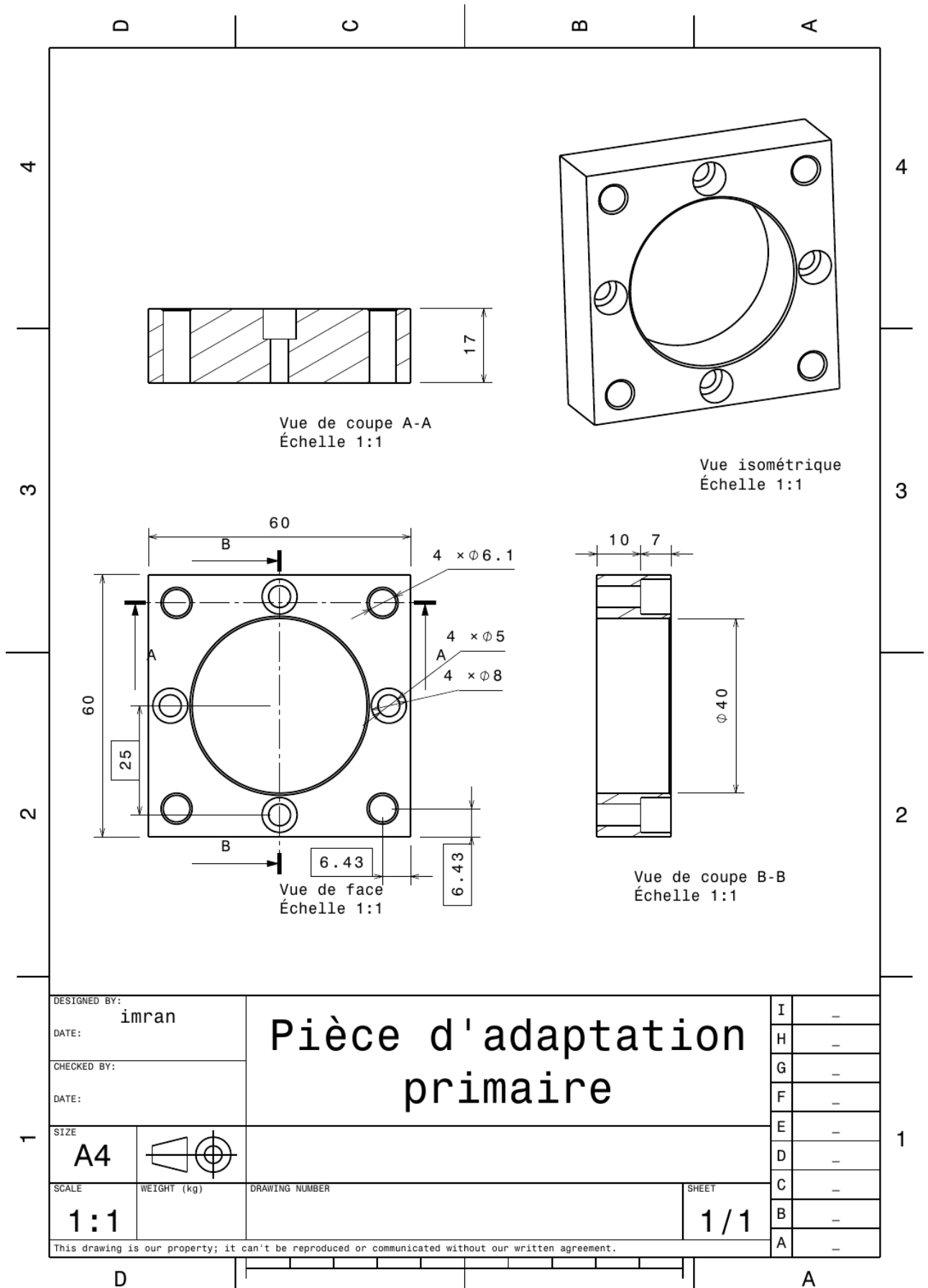


Figure 67 — Dessin de définition — Pièce d'adaptation primaire.

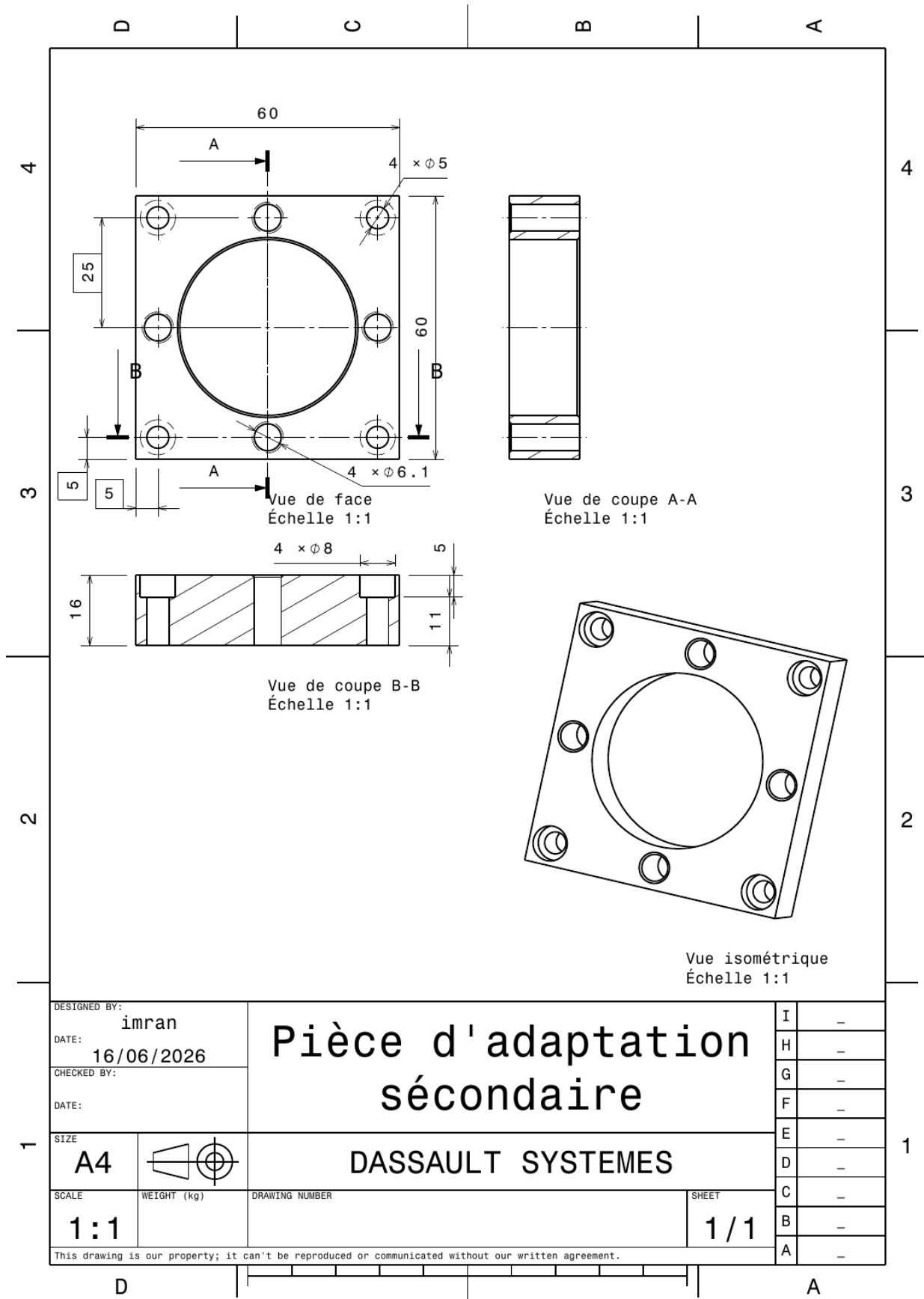


Figure 68 — Dessin de définition — Pièce d'adaptation secondaire.

Annexe C — Programme Arduino (firmware du banc)

Firmware de pilotage du moteur et de lecture de la cellule de charge (Arduino + powerSTEP01 + HX711).

```
/*
 * moteur_complet_hx711_v4.ino
 *
 * Historique :
 * v1 -> v2 : TARE corrigé (get_value au lieu de read), commande STREAM,
 *          homing en 3 temps avec décalage de 148 mm.
 * v3      : arrêt automatique de l'essai à une distance fixe.
 *
 * Nouveautés v4 :
 * (1) ARRÊT A 6,0 mm (au lieu de 6,2 mm). 6,0 mm = 15060 µsteps à
 *     2510 µsteps/mm.
 *
 * (2) ECHANTILLONNAGE PENDANT LE PALIER ("dwell sampling").
 *     Problème réglé : avant, le firmware ne diffusait des données que
 *     PENDANT le mouvement du moteur (état run mesure), puis se taisait
 *     pendant les 90 s de palier. Or :
 *     - les échantillons pris pendant le mouvement sont parasités par
 *       les transitoires électriques du moteur (valeurs en #millions) ;
 *     - la force STABILISÉE pendant le palier (ce qu'on veut pour E)
 *       n'était jamais enregistrée.
 *     Correction : en run mesure, quand le mouvement du palier se termine,
 *     le firmware NE repasse PLUS en standby : il continue de diffuser
 *     pendant tout le palier, moteur à l'arrêt (donc signal propre).
 *     Le palier suivant (RUNMESURE) relance simplement un mouvement.
 *     -> Le script MATLAB n'a pas besoin de changer : il détecte les
 *     paliers via la vitesse de la position (colonne 2), ignore la
 *     première moitié (le mouvement, bruité) et moyenne la seconde
 *     moitié (le palier stabilisé, propre).
 *
 * (3) HORLOGE D'ESSAI CONTINUE. startTime n'est réglé qu'au PREMIER
 *     RUNMESURE de l'essai ; les paliers suivants ne le remettent pas à
 *     zéro. Le timestamp (colonne 1) est donc monotone sur tout l'essai,
 *     indispensable pour le calcul de vitesse (diff(time)) côté MATLAB.
 *
 * NB : pendant le mouvement on enregistre quand même des lignes (bruitées) ;
 *     c'est normal et sans conséquence, MATLAB les jette (1ère moitié).
 *
 * Librairie requise : "HX711 Arduino Library" par Bogdan Necula (Bogde).
 */

#include <SPI.h>
#include <Ponoor_PowerSTEP01Library.h>
#include <HX711.h>

// =====
const int PIN_NCS      = 10;
const int PIN_STBY_RESET = 8;
const int LIMIT_LEFT_PIN = 5;
const int LIMIT_RIGHT_PIN = 6;
const int HX711_DT_PIN  = 7;
const int HX711_SCK_PIN  = 3;

powerSTEP driver(0, PIN_NCS, PIN_STBY_RESET);
HX711 cell;

const float V_SUPPLY = 24.0;
const float V_MOTOR  = 1.8;
const int VIT_HOMING = 400;           // vitesse vers la butée DROITE
const int VIT_RETOUT = 200;           // vitesse de dégagement de la butée
const int VIT_HOMING_OFFSET = 300;    // vitesse du déplacement de 148 mm
const long HOMING_OFFSET_USTEPS = 148L * 2510L; // 148 mm vers la GAUCHE (= 371480 µsteps)

// --- Arrêt automatique de l'essai à 6,0 mm -----
// 2510 µsteps/mm (1/32 microstepping, mesuré au banc). 6,0 mm = 15060 µsteps.
const long ECHELLE_USTEPS_PAR_MM = 2510L;
const long LIMITE_DEPL_USTEPS = 6L * ECHELLE_USTEPS_PAR_MM; // 6,0 mm = 15060
long posEssaiStart = 0; // position moteur au début de l'essai
bool referenceEssaiDefinie = false; // vrai dès le 1er RUNMESURE d'un essai
bool limiteAtteinte = false; // vrai quand 6,0 mm atteint -> bloque les mouvements

// Machine à états
enum EtatMachine { ETAT_STANDBY, ETAT_HOMING, ETAT_HOMING_BACKOFF, ETAT_HOMING_OFFSET, ETAT_RETOUT_GAUCHE,
                  ETAT_RETOUT_DROITE, ETAT_RUN, ETAT_RUN_MESURE, ETAT_STREAM };
EtatMachine etat = ETAT_STANDBY;

unsigned long startTime = 0;
unsigned long lastSend = 0;
bool modeInfini = false;
long currentDirection = 0;
bool stopRequested = false;

long forceRaw = 0; // valeur courante (offset par tare déjà soustrait)

// =====
void gererSerie();
void demarrerMouvementInfini(float vitesse, int sens);
void demarrerMouvementRelatif(float vitesse, int sens, long pas);
void stopMoteur();
bool gauche_atteinte();
bool droite_atteinte();
void mettreAJourForce();

// =====
void setup() {
    Serial.begin(115200);
```

```
Serial.println("init...");

pinMode(PIN_STBY_RESET, OUTPUT);
pinMode(PIN_NCS, OUTPUT);
digitalWrite(PIN_NCS, HIGH);
pinMode(LIMIT_LEFT_PIN, INPUT_PULLUP);
pinMode(LIMIT_RIGHT_PIN, INPUT_PULLUP);

digitalWrite(PIN_STBY_RESET, LOW);
delay(10);
digitalWrite(PIN_STBY_RESET, HIGH);
delay(10);

SPI.begin();
SPI.setDataMode(SPI_MODE3);
driver.SPIPortConnect(&SPI);

driver.configStepMode(STEP_FS_32);
driver.setMaxSpeed(1000);
driver.setMinSpeed(0);
driver.setAcc(2000);
driver.setDec(2000);
driver.setSlewRate(SR_520V_us);
driver.setOCThreshold(8);
driver.setOCShutdown(OC_SD_ENABLE);
driver.setPWMFreq(PWM_DIV_1, PWM_MUL_0_75);
driver.setVoltageComp(VS_COMP_DISABLE);
driver.setSwitchMode(SW_HARD_STOP);
driver.setOscMode(INT_16MHZ);

uint8_t kval = (uint8_t)((V_MOTOR / V_SUPPLY) * 255.0);
driver.setRunKVAL(kval);
driver.setAccKVAL(kval);
driver.setDecKVAL(kval);
driver.setHoldKVAL(kval / 2);

driver.setParam(ALARM_EN, 0x8F);
driver.getStatus();

cell.begin(HX711_DT_PIN, HX711_SCK_PIN);
// offset interne = 0 au démarrage. Un TARE explicite est nécessaire
// pour soustraire le zéro mécanique (le simple poids du montage).

startTime = millis();
Serial.println("ready");
}

// =====
void loop() {
    mettreAJourForce();

    switch (etat) {

        case ETAT_STANDBY:
            gererSerie();
            break;

        case ETAT_HOMING:
            gererSerie();
            // On va vers la butée DROITE.
            if (droite_atteinte()) {
                driver.hardStop();
                Serial.println("homing backoff");
                etat = ETAT_HOMING_BACKOFF;
                delay(10);
            }
            break;

        case ETAT_HOMING_BACKOFF:
            gererSerie();
            // On dégage la butée droite en reculant vers la GAUCHE (REV).
            if (droite_atteinte()) {
                if (driver.busyCheck() == 0) {
                    driver.run(REV, VIT_RETOUR);
                }
            } else {
                driver.softStop();
                while (driver.busyCheck());
                // Capteur relâché : on lance le déplacement de 148 mm vers la GAUCHE.
                driver.setMaxSpeed(VIT_HOMING_OFFSET);
                driver.move(REV, HOMING_OFFSET_USTEPS);
                etat = ETAT_HOMING_OFFSET;
                Serial.println("homing offset");
            }
            break;

        case ETAT_HOMING_OFFSET:
            gererSerie();
            // Sécurité : si on touche la butée GAUCHE pendant le déplacement
            // (148 mm dépasse la course dispo), on stoppe net.
            if (gauche_atteinte()) {
                driver.hardStop();
                while (driver.busyCheck());
                etat = ETAT_STANDBY;
                Serial.println("standby");
                break;
            }
            if (driver.busyCheck() == 0) { // déplacement terminé
                etat = ETAT_STANDBY;
                Serial.println("standby");
            }
            break;
    }
}
```

```

case ETAT RETOUR_DROITE:
gererSerie();
if (gauche_atteinte()) {
if (driver.busyCheck() == 0) {
driver.run(FWD, VIT_RETOUR);
}
} else {
driver.softStop();
while (driver.busyCheck());
etat = ETAT_STANDBY;
Serial.println("standby");
}
break;

case ETAT RETOUR_GAUCHE:
gererSerie();
if (droite_atteinte()) {
if (driver.busyCheck() == 0) {
driver.run(REV, VIT_RETOUR);
}
} else {
driver.softStop();
while (driver.busyCheck());
etat = ETAT_STANDBY;
Serial.println("standby");
}
break;

case ETAT RUN:
gererSerie();

if (gauche_atteinte()) {
driver.hardStop();
Serial.println("retour droite");
etat = ETAT_RETOUR_DROITE;
delay(10);
break;
}
if (droite_atteinte()) {
driver.hardStop();
Serial.println("retour gauche");
etat = ETAT_RETOUR_GAUCHE;
delay(10);
break;
}

// Le jog (RUN) revient en standby à la fin du mouvement, comme avant.
if (!modeInfini && driver.busyCheck() == 0 && !stopRequested) {
etat = ETAT_STANDBY;
Serial.println("standby");
}
if (driver.busyCheck() != 0) stopRequested = false;
break;

case ETAT RUN_MESURE:
gererSerie();

if (gauche_atteinte()) {
driver.hardStop();
Serial.println("retour droite");
etat = ETAT_RETOUR_DROITE;
delay(10);
break;
}
if (droite_atteinte()) {
driver.hardStop();
Serial.println("retour gauche");
etat = ETAT_RETOUR_GAUCHE;
delay(10);
break;
}

// --- Arrêt automatique à 6,0 mm de déplacement cumulé -----
if (referenceEssaiDefinie) {
long depl = driver.getPos() - posEssaiStart;
if (depl < 0) depl = -depl; // indépendant du sens
if (depl >= LIMITE_DEPL_USTEPS) {
driver.hardStop();
while (driver.busyCheck());
limiteAtteinte = true; // bloque les mouvements suivants
etat = ETAT_STANDBY;
Serial.println("limite 6.0mm");
Serial.println("standby");
break;
}
}

// DWELL SAMPLING : on NE repasse PAS en standby à la fin du mouvement.
// On continue de diffuser pendant tout le palier (moteur à l'arrêt
// = signal propre). Le palier suivant relancera un mouvement.
// (plus de transition "busyCheck()==0 -> standby" ici, volontairement)

if (millis() - lastSend > 100) {
lastSend = millis();
long t = millis() - startTime;
Serial.print(t);
Serial.print(driver.getPos());
Serial.print(',');
Serial.print(0);
Serial.println(forceRaw);
}
break;

case ETAT STREAM:

```

```

    gererSerie();

    // pas de gestion fin de course : on ne bouge pas le moteur.
    // Format CSV identique au RUN MESURE pour ne pas perturber Node-RED :
    // time, 0 (pas de moteur), 0 (réservé encodeur), forceRaw
    if (millis() - lastSend > 100) {
        lastSend = millis();
        long t = millis() - startTime;
        Serial.print(t);          Serial.print(',');
        Serial.print(0);          Serial.print(',');
        Serial.print(0);          Serial.print(',');
        Serial.println(forceRaw);
    }
    break;
}
}

// =====
void gererSerie() {
    if (!Serial.available()) return;
    String cmd = Serial.readStringUntil('\n');
    cmd.trim();

    if (cmd == "STOP") {
        stopMoteur();
        etat = ETAT_STANDBY;
        stopRequested = true;
        // Fin d'essai : on réarme la détection de référence et on débloque.
        referenceEssaiDefinie = false;
        limiteAtteinte = false;
        Serial.println("standby");
        while (Serial.available()) Serial.read();
        return;
    }

    if (cmd == "TARE") {
        if (cell.wait_ready_timeout(1000)) {
            cell.tare(10); // moyenne 10 lectures, stocke l'offset interne
            forceRaw = 0;
            Serial.println("tare ok");
        } else {
            Serial.println("tare: HX711 non détectée");
        }
        return;
    }

    if (cmd == "STREAM") {
        // diffusion continue de la force sans bouger le moteur
        startTime = millis();
        lastSend = 0;
        etat = ETAT_STREAM;
        Serial.println("stream");
        return;
    }

    if (cmd == "HOME") {
        driver.run(FWD, VIT_HOMING); // vers la butée DROITE
        etat = ETAT_HOMING;
        Serial.println("homing");
    }
    else if (cmd.startsWith("RUN:") || cmd.startsWith("RUNMESURE:")) {
        bool mesure = cmd.startsWith("RUNMESURE:");

        // Si la limite de 6,0 mm a déjà été atteinte, on ignore tout mouvement
        // tant qu'un STOP n'a pas été reçu (réarmement de l'essai).
        if (limiteAtteinte) {
            Serial.println("limite atteinte - envoyer STOP");
            return;
        }

        int p1 = cmd.indexOf(':');
        int p2 = cmd.indexOf(',', p1 + 1);
        int p3 = cmd.indexOf(',', p2 + 1);
        if (p1 == -1 || p2 == -1 || p3 == -1) {
            Serial.println("commande invalide");
            return;
        }
        int vitesse = cmd.substring(p1 + 1, p2).toInt();
        int sens = cmd.substring(p2 + 1, p3).toInt();
        long modeOuPas = cmd.substring(p3 + 1).toInt();

        if (mesure) {
            // Premier palier de l'essai : on fixe la référence ET l'horloge.
            // Paliers suivants : on NE touche PAS startTime (timestamp continu).
            if (!referenceEssaiDefinie) {
                posEssaiStart = driver.getPos();
                referenceEssaiDefinie = true;
                startTime = millis();
                lastSend = 0;
            }
        }
        else {
            // Jog manuel : horloge locale à chaque commande.
            startTime = millis();
        }

        if (modeOuPas == -1)
            demarrerMouvementInfini(vitesse, sens);
        else
            demarrerMouvementRelatif(vitesse, sens, modeOuPas);

        etat = mesure ? ETAT_RUN_MESURE : ETAT_RUN;
        Serial.println(mesure ? "run mesure" : "run");
    }
}

```

```
// =====
void demarrerMouvementInfini(float vitesse, int sens) {
    vitesse = abs(vitesse);
    driver.setMaxSpeed(vitesse);
    driver.run((sens >= 0) ? FWD : REV, vitesse);
    modeInfini = true;
    currentDirection = (sens >= 0) ? 1 : -1;
    stopRequested = false;
}

void demarrerMouvementRelatif(float vitesse, int sens, long pas) {
    vitesse = abs(vitesse);
    driver.setMaxSpeed(vitesse);
    driver.move((sens >= 0) ? FWD : REV, pas);
    modeInfini = false;
    currentDirection = (sens >= 0) ? 1 : -1;
    stopRequested = false;
}

void stopMoteur() {
    driver.hardStop();
    while (driver.busyCheck());
    modeInfini = false;
    stopRequested = true;
}

// =====
bool gauche_atteinte() { return digitalRead(LIMIT_LEFT_PIN) == HIGH; }
bool droite_atteinte() { return digitalRead(LIMIT_RIGHT_PIN) == HIGH; }

// =====
// Met à jour forceRaw avec la valeur courante de la cellule, offset
// (= tare) SOUSTRAIT. Si aucune nouvelle lecture n'est dispo, on garde
// la valeur précédente. get_value() applique l'offset stocké par tare().
void mettreAJourForce() {
    if (cell.is_ready()) {
        forceRaw = cell.get_value();
    }
}
```

Annexe D — Flux Node-RED

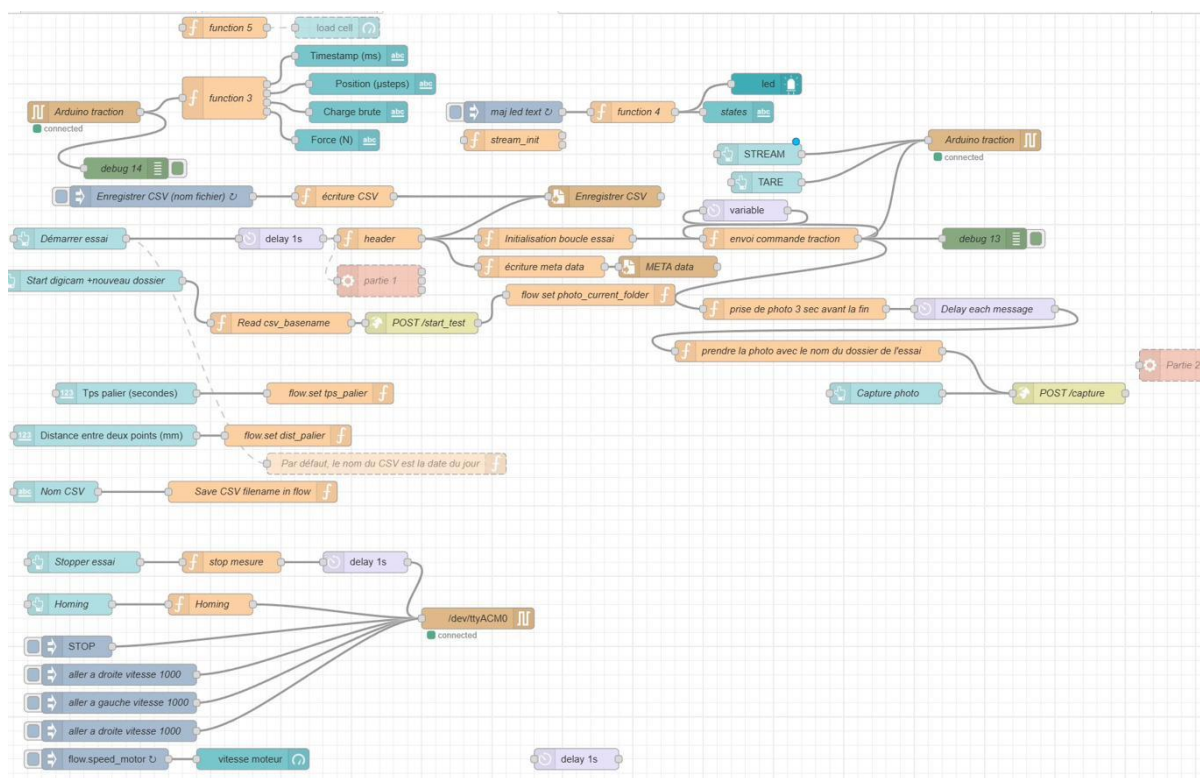


Figure 69 — Flux Node-RED de pilotage du banc et d'acquisition des données.

Annexe E — Scripts MATLAB de traitement des données

Chaîne de traitement sous MATLAB (Image Processing Toolbox) décrite au § 3.3.9. Les scripts sont reproduits dans l'ordre d'utilisation : calcul du module apparent, module corrigé par pointage manuel, puis tracés et agrégation des résultats. Constantes d'étalonnage : $1,5150 \times 10^{-3}$ mN/compte ; 2510 μ pas/mm ; échelle image 0,02239 mm/px.

Data_treatment_2026_06_10.m

Calcul du module de Young apparent à partir de la courbe force-déplacement : lecture robuste du CSV, détection des paliers, moyenne de la seconde moitié de chaque palier, puis pente sur la zone fixe (30 premiers paliers).

```
clear
close all
clc

% =====
% Data_treatment - corrected June 2026
%
% Changes vs Data treatment 2025_10_16.m:
% (1) FORCE: was `Data(:,4)*10` (no calibration at all). Now uses the
% new load cell's factor. New cell (June 2026): 0.00015443 g/count.
% force_mN = raw * 0.00015443 g/count * 9.81 mN/g = raw * 1.5150e-3
% Le compte brut devient POSITIF en traction sur cette cellule :
% pas de signe moins (tirer -> force positive).
% Sign: tensile produces NEGATIVE raw counts, so the minus makes
% tension come out positive.
% (2) POSITION: was `/1600*2` (= 800 steps/mm) - WRONG for this bench.
% The bench is 2510  $\mu$ steps/mm in 1/32 microstepping. Using 800
% under-counted displacement by ~3.14x and would have made E ~3x
% too large. Now divides by 2510.
% (3) ENCODER: column 3 is no longer populated (no encoder on this
% bench - the Arduino always emits 0 there). The old encoder plots
% used xlim/ylim([min max]) on an all-zero vector, which throws an
% error and halts the script before the analysis runs. Those plots
% are removed. The modulus analysis never used the encoder anyway.
%
% Palier-averaging logic and the .mat save are unchanged.
% =====

concentration = "0.3";
mass_sorbitol = "21.86g";

% --- Calibration constants -----
FORCE_MN_PER_RAW = 1.5150e-3; % mN per raw count. NB: raw count goes POSITIVE
% under tension on this cell, so NO minus sign
% (pulling -> positive force).
USTEPS_PER_MM = 2510; % 1/32 microstepping, measured at the bench

% base folder = 'G:\Traction parenchyme pomme de terre\BANC TOLE';
base_folder = 'C:\Users\imran\OneDrive\Documents\MATLAB\potato0,3mol';
folder_name = '0,3molEchantillon4';
path = [base_folder 'RAW DATA\' folder_name];
directoryContent = dir(path);

for i=1 : length(directoryContent)
    disp(directoryContent(i).name)
    if contains(directoryContent(i).name, '.csv')
        csv_name = directoryContent(i).name;
    end
end

% --- lecture ROBUSTE du CSV -----
% Tolere : (a) un en-tete present ou absent, (b) des lignes parasites
% AVANT le vrai en-tete (banc au repos avant le demarrage de l'essai,
% qui re-ecrit l'en-tete). On detecte la DERNIERE ligne "Timestamp(ms)"
% et on lit a partir de la ; sinon on lit tout en numerique.
fid = fopen([path csv_name]); raw_lines = textscan(fid,'%s','Delimiter','\n','Whitespace',''); fclose(fid);
raw_lines = raw_lines[1];
hdr_idx = find(contains(raw_lines,'Timestamp(ms)'), 1, 'last'); % derniere occurrence
if isempty(hdr_idx)
    Data = readmatrix([path csv_name]); % pas d'en-tete : tout numerique
else
    Data = readmatrix([path csv_name], 'NumHeaderLines', hdr_idx); % saute en-tete + parasites
end
Data = Data(all(~isnan(Data(:,1:2)),2), :); % vire d'eventuelles lignes vides/NaN

% CSV columns: 1 = timestamp(ms), 2 = motor position(usteps),
% 3 = unused (always 0, no encoder), 4 = raw HX711 count
% time = (Data(:,1)-Data(1,1))/1000 ; % s
% force = Data(:,4) * FORCE_MN_PER_RAW; % mN
displacement_moteur = (Data(:,2)-Data(1,2)) / USTEPS_PER_MM; % mm

% NOTE ON SIGN: the essai now pulls in the +1 (FWD) direction, so motor
% position increases during the pull and displacement_moteur is positive.
% If you ever see it come out negative (force-displacement curve in the
% wrong quadrant, or 0 paliers detected below), the pull direction was
% reversed - just put a leading minus back on the line above.

figure;
subplot(3,1,1)
plot(time, force);
xlabel('time (s)'); ylabel('force (mN)');
subplot(3,1,2)
plot(time, displacement_moteur)
```

```
xlabel('time (s)'); ylabel('displacement moteur (mm)');
subplot(3,1,3)
plot(displacement_moteur, force)
xlabel('displacement moteur (mm)'); ylabel('force (mN)');

% --- Palier detection (unchanged) -----
dev= (diff(displacement_moteur)./ diff(time)) > (0.1/1); % 0.1mm en 1 sec
ddev= diff(dev);
paliers=zeros(1, length(time));
paliers(1)=0;
for i=1:length(ddev)
    if ddev(i)>0
        paliers(i+1) = paliers(i)+1;
    else
        paliers(i+1) = paliers(i);
    end
end

end
paliers(end)=paliers(end-1);

for i=1:max(paliers)
    force_palier = force(paliers==i);
    displacement_palier = displacement_moteur(paliers==i);
    force_palier= force_palier (floor(length(force_palier)/2):end);
    displacement_palier = displacement_palier(floor(length(force_palier)/2):end);
    MeanForce(i)= mean(force_palier);
    Mean_disp(i)= mean(displacement_palier);
    TempsPalier(i) = mean(time(paliers==i));
end

figure;
plot(Mean_disp, MeanForce, 'r')
xlabel('Mean\ disp (mm)')
ylabel('MeanForce (mN)')

%% =====
% MODULE DE YOUNG E = sigma / epsilon
% E = (pente force-deplacement) * L0 / A
% Force en mN, deplacement et L0 en mm, A en mm^2 -> E en kPa
% (1 mN/mm^2 = 1 kPa)
% -> EDITER L0 et d pour CHAQUE essai.
% =====
L0 = 17.02; % longueur utile = BLANC EXPOSE entre mors, mm (pied a coulisse)
% (PAS les 30.98 mm mors-a-mors : la partie serree ne se deforme pas)
d = 4.612; % diametre echantillon GONFLE (apres trempage), mm
% ~4.92 mesure sur l'image ; coherent avec 4.85-5.1 au pied a coulisse.
% -> coherent avec le diametre mesure au pied a coulisse.
A = pi*(d/2)^2; % section, mm^2 (cylindre)

% Region lineaire utilisee pour la pente : indices des paliers.
% Par default la 1ere moitie. REGARDER la figure et ajuster si besoin,
% p.ex. idx = 1:15; pour ne garder que le debut rectiligne.
idx = 1:round(numel(Mean_disp)/2);

pfit = polyfit(Mean_disp(idx), MeanForce(idx), 1);
slope = pfit(1); % mN/mm
E_kPa = slope * L0 / A; % kPa

% droite ajustee superposee sur la courbe
hold on
plot(Mean_disp(idx), polyval(pfit, Mean_disp(idx)), '-k', 'LineWidth', 1.5)
hold off
title(sprintf('pente = %.2f mN/mm -> E = %.1f kPa (%.3f MPa)', ...
    slope, E_kPa, E_kPa/1000))

fprintf('\n=== Module de Young ===\n');
fprintf('L0 = %.2f mm | d = %.2f mm | A = %.2f mm^2\n', L0, d, A);
fprintf('points utilisees (lineaire) : %d a %d sur %d paliers\n', ...
    idx(1), idx(end), numel(Mean_disp));
fprintf('pente = %.3f mN/mm\n', slope);
fprintf('E = %.1f kPa (= %.4f MPa)\n', E_kPa, E_kPa/1000);

%% =====
% COURBE CONTRAINTE-DEFORMATION (sigma vs epsilon) -- graphe pour M. Tadrast
% sigma = Force / A0 (MPa); epsilon = deplacement / L0 (sans unite)
% NB : epsilon ici = deformation MOTEUR (crosshead). Le glissement aux
% mors la surestime ; E corrige reel = via Verif_manuelle_multi (clicks).
% =====
epsilon = Mean_disp / L0; % deformation (motor/crosshead)
sigma = (MeanForce/1000) / A; % MPa (mN->N, /mm^2)

pE = polyfit(epsilon(idx), sigma(idx), 1);
Emod = pE(1); % MPa = pente sigma/epsilon

figure;
plot(epsilon, sigma, 'r'); hold on;
plot(epsilon(idx), polyval(pE, epsilon(idx)), '-k', 'LineWidth', 1.5); hold off;
xlabel('\epsilon (deformation)'); ylabel('sigma (MPa)');
title(sprintf('Contrainte-deformation 0 mol/L | E = %.3f MPa (apparent, moteur)', Emod));
grid on;

fprintf('\n=== Courbe sigma-epsilon ===\n');
fprintf('E (pente sigma/epsilon, deformation moteur) = %.3f MPa\n', Emod);
fprintf(' -> APPARENT (course moteur). E corrige = via Verif_manuelle_multi (clicks).\n');

%% =====
% SAUVEGARDE (E APPARENT seulement)
% Le E corrige du glissement n'est PLUS calcule ici : il provient
% desormais du pointage MANUEL des points (Verif_manuelle_multi.m),
% bien plus fiable que l'ancienne detection automatique d'images.
% -> Ce script ne fait qu'une chose : le E APPARENT (force vs moteur).
% =====
```

```
save([base folder '\META DATA TOLE\' folder name 'force_temps.mat'], ...  
      "Mean disp", "MeanForce", "time", "TempsPalier", "force", ...  
      "concentration", "mass sorbitol", ...  
      "L0", "d", "A", "slope", "E_kPa", '-mat')
```

Verif_manuelle_multi.m

Module corrigé par pointage manuel : sur six images réparties sur les paliers, l'opérateur clique les deux points marqués sur l'échantillon ; la pente force-allongement réel donne E corrigé, insensible au glissement aux mors.

```
clear; close all; clc;

% =====
% Verif_manuelle_multi.m (suivi de points MANUEL, multi-images)
%
% Mesure l'allongement REEL en cliquant les 2 points sur 6 images
% réparties sur les paliers ~1.30, puis ajuste la pente force vs
% allongement-image -> E corrige (même région que E apparent).
% Insensible au fond clair / aux ombres de mors (c'est VOUS qui visez).
%
% -> Ajoute UNE ligne par essai dans RESULTATS_manuel.xlsx.
% Relancer le même essai ECRASE sa ligne (pas de doublon).
%
% POUR CHAQUE ESSAI :
% 1) éditer base folder, folder_name, L0, d (puis Run)
% 2) 6 images s'ouvrent l'une après l'autre :
%     clic 1 = point du HAUT, clic 2 = point du BAS
%     (zoom molette possible AVANT de cliquer)
% 3) un graphe de contrôle s'affiche (force vs allongement) ; si un
%     point est aberrant (clic rate) -> relancer cet essai.
%
% Lit force_temps.mat s'il existe (force/déplacement/L0/A) ; sinon
% retombe sur le CSV brut. Pas besoin de relancer Data_treatment.
% =====

% ----- A EDITER POUR CHAQUE ESSAI -----
base_folder = 'C:\Users\imran\OneDrive\Documents\MATLAB\potato\0,3mol';
folder_name = '0,3molEchantillon4';
L0 = 17.02; % mm (pied à coulisse) -- utilise si pas dans le .mat
d = 4.612; % mm
scale = 0.02239; % mm/pixel
n_frames_click = 6; % nb d'images à cliquer (défaut 6)
fit_palier_max = 30; % région d'ajustement (paliers 1..30, comme E apparent)
results_xlsx = 'C:\Users\imran\OneDrive\Documents\MATLAB\potato\RESULTATS_manuel.xlsx';
% -----

USTEPS_PER_MM = 2510; FORCE_MN_PER_RAW = 1.5150e-3;
sample = erase(folder_name, '\');
conc = str2double(strrep(strrep(regexprep(folder_name, 'Echantillon.*', ''), 'mol', ''), ', ', '.'));

% ---- 1) charger force / déplacement moteur / temps des paliers ----
matf = [base_folder 'META_DATA_TOLE\' folder_name 'force_temps.mat'];
if isfile(matf)
    F = load(matf);
    MeanForce = F.MeanForce(:); Mean_disp = F.Mean_disp(:);
    TempsPalier = F.TempsPalier(:);
    good = isfinite(TempsPalier) & isfinite(MeanForce) & isfinite(Mean_disp);
    TempsPalier = TempsPalier(good); MeanForce = MeanForce(good); Mean_disp = Mean_disp(good);
    if isfield(F, 'L0'), L0 = F.L0; end
    if isfield(F, 'A'), A = F.A; else, A = pi*(d/2)^2; end
    fprintf('force_temps.mat charge (%d paliers).\n', numel(MeanForce));
else
    % --- fallback : relire le CSV brut et refaire les paliers ---
    A = pi*(d/2)^2;
    path = [base_folder 'RAW_DATA\' folder_name];
    dc = dir([path '*.csv']); if isempty(dc), error('Pas de CSV dans %s', path); end
    D = readmatrix([path dc(1).name]);
    time = (D(:,1)-D(1,1))/1000; force = D(:,4)*FORCE_MN_PER_RAW;
    disp_m = (D(:,2)-D(1,2))/USTEPS_PER_MM;
    dev = (diff(disp_m)./diff(time)) > 0.1; ddev = diff(dev);
    pal = zeros(1, numel(time));
    for i=1: numel(ddev), pal(i+1) = pal(i) + (ddev(i)>0); end
    pal(end)=pal(end-1);
    MeanForce=[]; Mean_disp=[]; TempsPalier=[];
    for i=1: max(pal)
        fp=force(pal==i); dp=disp_m(pal==i); tp=time(pal==i);
        fp=fp(floor(numel(fp)/2):end); dp=dp(floor(numel(dp)/2):end);
        MeanForce(i)=mean(fp); Mean_disp(i)=mean(dp); TempsPalier(i)=mean(tp); %ok<AGROW>
    end
    MeanForce=MeanForce(:); Mean_disp=Mean_disp(:); TempsPalier=TempsPalier(:);
    fprintf('CSV brut relu (%d paliers) ; L0=%2f d=%2f saisis.\n', numel(MeanForce), L0, d);
end
npal = numel(MeanForce);
ne = min(fit_palier_max, npal);

% ---- 2) lister les images et choisir n_frames réparties 1..ne ----
path = [base_folder 'RAW_DATA\' folder_name];
files = dir([path '*.jpg']);
if isempty(files), error('Aucune image dans %s', path); end
[~,o] = sort([files.name]); files = files(o);
% temps (s) de chaque image d'après le nom HHMMSS
timg = arrayfun(@(f) str2double(f.name(1:2))*3600 + str2double(f.name(3:4))*60 + ...
    str2double(f.name(5:6)), files);
timg = timg - timg(1);
tpal = TempsPalier - TempsPalier(1);

% paliers cibles repartis sur 1..ne, puis image la plus proche en temps
target_pal = round(linspace(1, ne, n_frames_click));
sel = zeros(n_frames_click, 1);
for i=1: n_frames_click
    [~,sel(i)] = min(abs(timg - tpal(target_pal(i))));
end
sel = unique(sel, 'stable');

% ---- 3) clics : 2 points par image ----
dot_mm = nan(numel(sel), 1); fpal = nan(numel(sel), 1); dmot = nan(numel(sel), 1);
for i=1: numel(sel)
    I = imread([path files(sel(i)).name]);
    fig = figure('Name', sprintf('Image %d/%d (palier ~%d)', i, numel(sel), target_pal(i)), 'NumberTitle', 'off');
    imshow(I); hold on;
```

```

title(sprintf('%s - image %d/%d', sample, i, numel(sel)), ...
    'Zoom librement (loupe/molette). Cliquez le point HAUT, puis le point BAS.'). 'Interpreter','none');
p1 = drawpoint('Color','g'); % cliquez le point HAUT (zoom/pan possible avant)
p2 = drawpoint('Color','g'); % cliquez le point BAS
P1 = p1.Position; P2 = p2.Position;
plot([P1(1) P2(1)], [P1(2) P2(2)], 'g-'); drawnow;
dot_mm(i) = hypot(P2(1)-P1(1), P2(2)-P1(2)) * scale;
fpal(i) = interp1(tpal, MeanForce, timg(sel(i)), 'linear', 'extrap');
dmtot(i) = interp1(tpal, Mean_disp, timg(sel(i)), 'linear', 'extrap');
fprintf(' image %d : distance points = %.2f mm (force %.0f mN)\n', i, dot_mm(i), fpal(i));
end

% ---- 4) ajustement force vs allongement image -> E corrige -----
img_elong = dot_mm - dot_mm(1); % allongement reel
mot_elong = dmtot - dmtot(1); % course moteur
pim = polyfit(img_elong, fpal, 1); slope_true = pim(1);
E_corr_kPa = slope_true * L0 / A;
% E apparent (force vs moteur) sur les memes points cliques
pmo = polyfit(mot_elong, fpal, 1); E_app_kPa = pmo(1) * L0 / A;
loss = 1 - (img_elong(end)-img_elong(1)) / (mot_elong(end)-mot_elong(1));
eps_ax = (dot_mm(end)-dot_mm(1)) / dot_mm(1);

% ---- 5) graphe de controle -----
figure('Name','Contrôle : force vs allongement','NumberTitle','off');
plot(mot_elong, fpal, 'r+', 'MarkerSize',10, 'LineWidth',1.5); hold on;
plot(img_elong, fpal, 'bo', 'MarkerSize',8, 'LineWidth',1.5);
plot(img_elong, polyval(pim, img_elong), 'b-', 'LineWidth',1.5);
grid on; xlabel('allongement (mm)'); ylabel('force (mN)');
legend({'moteur (apparent)', 'points images (reel)', 'fit image'}, 'Location','northwest');
title(sprintf('%s : E {app} %.3f -> E {corr} %.3f MPa (perte course %.0f%%)', ...
    sample, E_app_kPa/1000, E_corr_kPa/1000, 100*loss), 'Interpreter','none');

fprintf('\n=== %s (MANUEL, %d images) ===\n', sample, numel(sel));
fprintf('E apparent = %.3f MPa | E corrige = %.3f MPa | perte course = %.0f%%\n', ...
    E_app_kPa/1000, E_corr_kPa/1000, 100*loss);
fprintf('allongement image total = %.2f mm | deformation axiale = %.3f\n', ...
    img_elong(end), eps_ax);
if any(abs(img_elong - polyval(polyfit(fpal, img_elong, 1), fpal)) > 0.3*max(abs(img_elong)))
    warning('Un point semble aberrant (clic rate ?) -> verifier le graphe, relancer si besoin.');
```

```

end

% ---- 6) ajouter / ecraser la ligne dans le tableau resultats -----
newrow = table(string(sample), conc, L0, d, A, ...
    round(E_app_kPa/1000,3), round(E_corr_kPa/1000,3), round(100*loss,0), round(eps_ax,3), ...
    'VariableNames',{'echantillon','concentration_molL','L0_mm','d_mm','A_mm2', ...
    'E_app_MPa','E_corr_manuel_MPa','perte_course_pct','eps_axial'});
if isfile(results_xlsx)
    T = readtable(results_xlsx, 'TextType','string');
    T(strcmp(T.echantillon, sample), :) = []; % enlever ancienne ligne meme essai
    T = [T; newrow];
else
    T = newrow;
end
T = sortrows(T, {'concentration_molL', 'echantillon'});
writetable(T, results_xlsx);
fprintf('-> ligne ajoutée dans %s (%d essais au total)\n', results_xlsx, height(T));

```

Plot_sigma_epsilon_v2.m

Tracé des courbes contrainte-déformation (σ - ϵ) : tous les essais retenus puis concentration par concentration.

```
clear; close all; clc;
% =====
% Plot_sigma_epsilon.m
%
% Courbes contrainte-déformation (APPARENT, deformation moteur).
% Les courbes corrigees-image sont RETIREES.
%
% RE-ZERO de la deformation : chaque essai a un offset de deplacement
% arbitraire au depart (rattrapage de jeu, pre-tension, logging a vide,
% voire recul moteur -> epsilon negatif). On re-zero epsilon au DEBUT
% REEL du chargement (1er palier ou la force depasse un seuil), de sorte
% que toutes les courbes partent de epsilon ~ 0. Cela ne change PAS la
% pente (donc E inchange) : c'est purement un recalage de l'axe x.
%
% TROIS niveaux de figures :
% (1) PAR ECHANTILLON -> ...\

```

Plot_E_vs_concentration_v2.m

Tracé du module de Young (apparent et corrigé) en fonction de la concentration en sorbitol.

```
clear; close all; clc;
% =====
% Plot_E_vs_concentration.m
%
% Courbe E (apparent ET corrige) en fonction de la concentration.
% - E apparent : lu dans chaque force temps.mat (Data treatment)
% - E corrige : lu dans RESULTATS_manuel.xlsx (pointage MANUEL,
%               Verif_manuelle_multi) -- SOURCE UNIQUE du corrige.
%
% Graphe AGREGÉ -> sauvegarde a la RACINE (root).
% =====

root = 'C:\Users\imran\OneDrive\Documents\MATLAB\potato\';
xlsx = [root 'RESULTATS_manuel.xlsx'];

% --- exclusions (nom d'echantillon EXACT) -----
exclude_app = {};
exclude_corr = {'0,2molEchantillon4'}; % corrige aberrant (perte 95%)

% --- 1) E APPARENT depuis les force temps.mat -----
conc_dirs = dir([root '*mol*']); conc_dirs = conc_dirs([conc_dirs.isdir]);
app = struct('name', {}, 'conc', {}, 'Eapp', {});
for k = 1:numel(conc_dirs)
    cdir = [root conc_dirs(k).name '\META DATA TOLE\'];
    subs = dir(cdir); subs = subs([subs.isdir] & ~startsWith(subs.name, '.'));
    for j = 1:numel(subs)
        mf = [cdir subs(j).name '\force_temps.mat'];
        if ~isfile(mf), continue; end
        S = load(mf); nm = subs(j).name;
        if any(strcmp(nm, exclude_app)), continue; end
        c = str2double(strrep(strrep(regexprep(nm, 'Echantillon.*', ''), 'mol', ''), ', ', ''));
        app(end+1) = struct('name', nm, 'conc', c, 'Eapp', S.E_kPa/1000); % #ok<SAGROW>
    end
end
if isempty(app), error('Aucun force_temps.mat trouve sous %s', root); end

% --- 2) E CORRIGE depuis RESULTATS_manuel.xlsx -----
T = readtable(xlsx, 'TextType', 'string');
ec = T.E_corr_manuel_MPa; ec(ec<=0) = NaN; % garde-fou physique
for i=1:numel(exclude_corr), ec(T.echantillon==string(exclude_corr{i})) = NaN; end

% --- 3) moyennes par concentration -----
ac = [app.conc]; ae = [app.Eapp]; cc = double(T.concentration_molL);
concs = unique([ac, cc, '']);
mEapp=nan(size(concs)); sEapp=nan(size(concs));
mEcor=nan(size(concs)); sEcor=nan(size(concs));
for i=1:numel(concs)
    a = ae(ac==concs(i));
    if ~isempty(a), mEapp(i)=mean(a); sEapp(i)=std(a); end
    e = ec(abs(cc-concs(i))<1e-9); e = e(~isnan(e));
    if ~isempty(e), mEcor(i)=mean(e); sEcor(i)=std(e); end
end

% --- 4) figure -----
f = figure('Color', 'w', 'Position', [100 100 760 520]);
yyaxis left
errorbar(concs, mEapp, sEapp, '-o', 'LineWidth', 1.6);
ylabel('E apparent (MPa)'); ylim([0 max(mEapp)*1.4]);
yyaxis right
errorbar(concs, mEcor, sEcor, '--s', 'LineWidth', 1.6);
ylabel('E corrige manuel (MPa)');
xlabel('concentration sorbitol (mol/L)');
title('E en fonction de la concentration (apparent vs corrige manuel)');
grid on; xlim([min(concs)-0.05 max(concs)+0.05]);
legend({'E apparent (moteur)', 'E corrige (pointage manuel)'}, 'Location', 'best');

fprintf('=== E vs concentration ===\n');
for i=1:numel(concs)
    fprintf(' c=%1f : E_app %3f +/- %3f | E_corr %2f +/- %2f MPa\n', ...
        concs(i), mEapp(i), sEapp(i), mEcor(i), sEcor(i));
end

% --- AGREGÉ -> racine ---
saveas(f, [root 'E_vs_concentration.png']);
fprintf('-> sauvegarde : %sE_vs_concentration.png\n', root);
```

Resume_resultats_v2.m

Agrégation des résultats : moyennes et écarts-types du module par concentration, tableau récapitulatif global.

```
clear; close all; clc;
% =====
% Resume_resultats.m
%
% Tableau recapitulatif : E apparent (force temps.mat) + E corrige
% manuel (RESULTATS manuel.xlsx), par echantillon et moyenne par
% concentration. EcFit RESUME global.xlsx a la RACINE.
% =====

root = 'C:\Users\imran\OneDrive\Documents\MATLAB\potato\';
xlsx = [root 'RESULTATS manuel.xlsx'];
exclude_corr = {'0,2molEchantillon4'};

% --- E apparent depuis les .mat -----
conc_dirs = dir([root '*mol']); conc_dirs = conc_dirs([conc_dirs.isdir]);
names=strings(0); concv=[]; Eapp=[]; L0v=[]; dv=[]; Av=[];
for k=1:numel(conc_dirs)
    cdir=[root conc_dirs(k).name '\META DATA TOLE\'];
    subs=dir(cdir); subs=subs([subs.isdir] & ~startsWith(subs.name, '.'));
    for j=1:numel(subs)
        mf=[cdir subs(j).name '\force temps.mat'];
        if ~isfile(mf), continue; end
        S=load(mf); nm=string(subs(j).name);
        c=str2double(strrep(strrep(regexprep(subs(j).name, 'Echantillon.*', ''), 'mol', ''), ', ', ''));
        names(end+1)=nm; concv(end+1)=c; Eapp(end+1)=S.E kPa/1000; %#ok<SAGROW>
        L0v(end+1)=S.L0; dv(end+1)=S.d; Av(end+1)=S.A; %#ok<SAGROW>
    end
end

% --- E corrige manuel depuis xlsx -----
M=readtable(xlsx, 'TextType', 'string');
Ecorr=nan(size(names)); loss=nan(size(names));
for i=1:numel(names)
    r=find(M.echantillon==names(i), 1);
    if ~isempty(r)
        Ecorr(i)=M.E_corr_manuel_MPa(r); loss(i)=M.perte_course_pct(r);
    end
end
Ecorr(Ecorr==0)=NaN;
for i=1:numel(exclude_corr), Ecorr(names==string(exclude_corr{i}))=NaN; end

% --- table par echantillon -----
Tu = table(names, 'concv', 'L0v', 'dv', 'Av', round(Eapp, 3), round(Ecorr, 3), loss, ' ...', 'VariableNames', {'echantillon', 'concentration molL', 'L0 mm', 'd mm', 'A mm2', ' ...', 'E app MPa', 'E corr manuel MPa', 'perte_course_pct'});
Tu = sortrows(Tu, {'concentration molL', '@chantillon'});
disp('--- par echantillon ---'); disp(Tu);

% --- moyennes par concentration -----
cs=unique(concv); rows=table();
for i=1:numel(cs)
    m=concv==cs(i);
    ea=Eapp(m); ec=Ecorr(m); ec=ec(~isnan(ec));
    rows=[rows; table(cs(i), numel(ea), round(mean(ea), 3), round(std(ea), 3), ...
        numel(ec), round(mean(ec), 3), round(std(ec), 3), ...
        'VariableNames', {'concentration molL', 'n app', 'E app moy', 'E app std', ...
            'n_corr', 'E_corr moy', 'E_corr std'})]; %#ok<AGROW>
end
disp('--- moyennes par concentration ---'); disp(rows);

writetable(Tu, [root 'RESUME_global.xlsx'], 'Sheet', 'par_echantillon');
writetable(rows, [root 'RESUME_global.xlsx'], 'Sheet', 'moyennes');
fprintf('-> %sRESUME_global.xlsx\n', root);
```

4^e de couverture

Champ	Contenu
Auteur	Imran BIN MOHD ZAKI
Formation	BUT Génie Mécanique et Productique, 2 ^e année — parcours CPD — IUT d'Aix-Marseille
Année	2025/2026 — soutenance le 22/06/2026
Lieu de stage	ISM (UMR 7287), équipe SBI — halle d'essai, IUT Aix-en-Provence
Tuteur entreprise	M. Loïc TADRIST
Tuteur universitaire	M. Eric JEANCELME
Logiciels	CATIA V5, MATLAB, Node-RED, Arduino IDE
Thème du stage	Conception et fabrication d'un banc de traction instrumenté pour caractériser le parenchyme de pomme de terre : mesure du module de Young et estimation du coefficient de Poisson en fonction de la concentration osmotique.
Mots-clés	parenchyme ; turgescence ; module de Young ; sorbitol ; banc d'essai
Nombre de pages du rapport	33 (hors annexes)
Nombre de pages des annexes	24 (Annexes A à E)

Résumé

Réalisé dans l'équipe Systèmes Bio-Inspirés de l'ISM, ce stage de 2^e année de BUT GMP avait pour but de caractériser le comportement mécanique du parenchyme de pomme de terre, modèle de solide cellulaire pressurisé. Comme le banc de référence était déjà utilisé, j'ai conçu un nouveau banc de traction (CATIA V5), je l'ai fabriqué (impression 3D PLA et résine, usinage aluminium) et instrumenté (cellule de charge HX711, acquisition par Raspberry Pi, pilotage Node-RED, arrêt d'urgence). Une campagne complète a été menée sur des solutions de sorbitol de 0 à 0,5 mol/L. Une chaîne MATLAB calcule le module de Young à partir de la courbe force-déplacement (module apparent) et de l'allongement réel mesuré sur les images (module corrigé). Le module apparent reste presque constant ($\approx 0,57\text{--}0,61$ MPa) jusqu'à 0,2 mol/L, puis devient environ 6 fois plus mou ($\approx 0,09$ MPa à 0,5 mol/L), comme l'explique le rôle de la turgescence. Le coefficient de Poisson, trop faible à mesurer avec une seule caméra, reste une piste d'amélioration.