

Simulation du capteur avec le logiciel LTSpice

Le logiciel LTSpice permet de simuler la réponse d'un circuit électrique à une stimulation. Nous proposons le schéma électrique suivant pour représenter notre capteur :

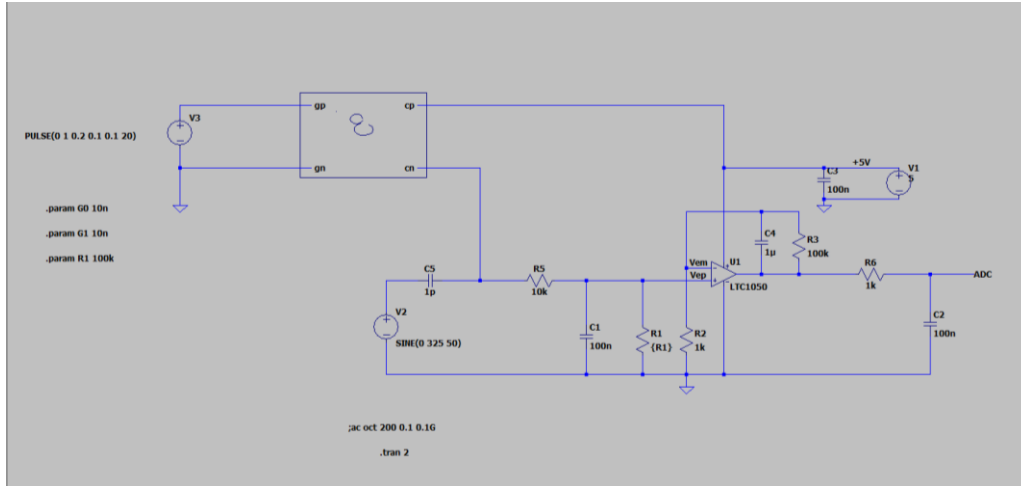


Figure 1. Circuit électrique du capteur de déformation à base de graphite

Ce circuit transimpédance convertit une variation de courant en variation de tension mesurable par la carte Arduino Uno utilisée dans le projet. Il est composé de 3 filtres afin de réduire le bruit lié à l'inertie du capteur, celui à 50 Hz et enfin diminuer le bruit ADC.

Fonctionnalité aux conditions nominales

La résistance du capteur étant de l'ordre du Giga Ohm et l'alimentation étant de 5 V, nous simulons la réponse du circuit à une stimulation de courant d'entrée de 100nA. La stimulation est un Pulse de 50nA à 100nA d'une durée de 2 secondes. Nous obtenons la réponse suivante :

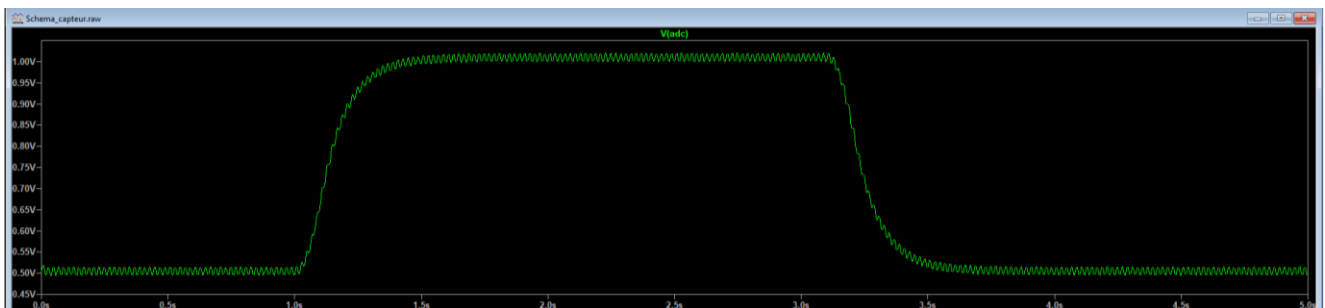


Figure 2. Réponse du capteur à un pulse d'intensité

Nous pouvons observer que le capteur est bien fonctionnel. En effet, la réponse suit la stimulation Pulse d'intensité et ne sature pas (plateau à $1V < 5V$).

Incidence de l'offset de l'amplificateur LTC1050C

D'après la datasheet du LTC1050C, l'offset typique d'entrée est de $0.5 \mu V$ et le gain de l'AOP est de 160 dB. Pour une tension d'entrée de 10mV, nous avons donc en sortie $10^6 V$. L'offset de sortie est quant à lui de 50V. L'offset est donc négligeable et son incidence est donc acceptable.

Incidence du courant d'entrée de l'amplificateur LTC1050C

Nous faisons varier I_{sens} et déterminons la valeur pour laquelle a lieu la saturation. La saturation en sortie vaut 4,8V et est atteinte pour un courant $I_{sens} = 480 \text{ nA}$.

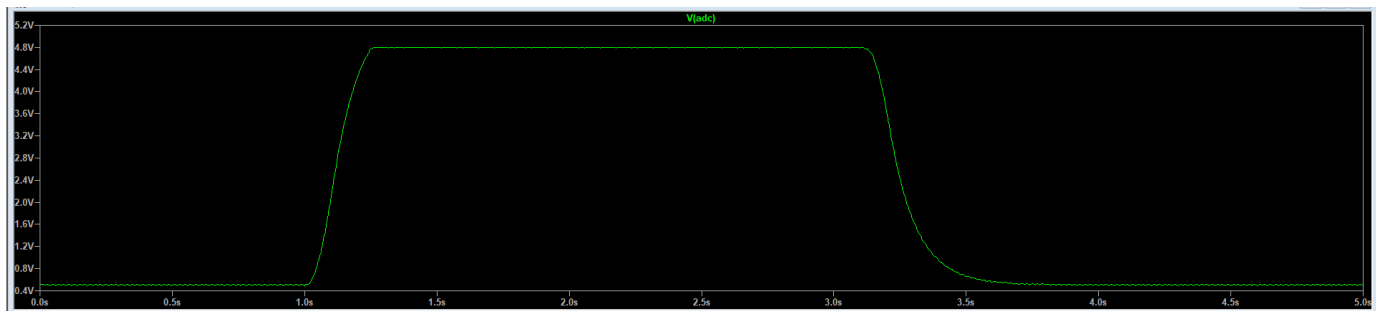


Figure 3. Saturation du capteur à un pulse d'intensité 480nA

Cette valeur de I_{sens} engendrant la saturation est liée à la valeur de la résistance R2. Dans cette simulation, R2 est fixe. Cependant, pour la réalisation réelle du capteur, nous avons décidé de remplacer R2 par un potentiomètre digital (= résistance variable) afin d'éviter les problèmes de saturation.

Atténuation globale d'un bruit en courant de 50 Hz

En simulant avec *LTSpice* le gain en fonction de la fréquence, nous obtenons la courbe ci-dessous :

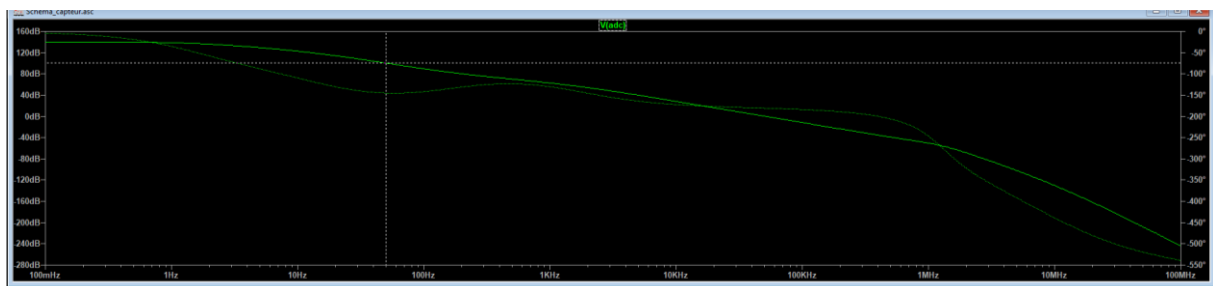


Figure 4. Balayage fréquentiel du gain du circuit

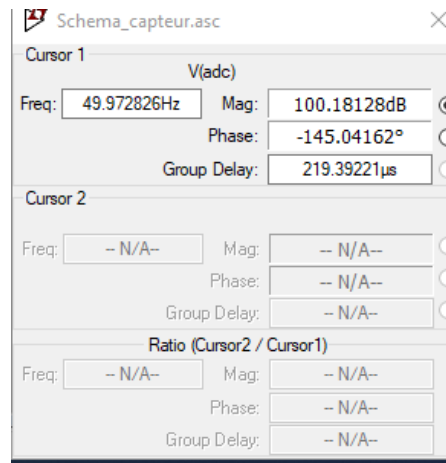


Figure 5. Résultat de l'analyse à 50Hz

L'analyse de la courbe permet de conclure que le bruit à 50Hz est atténué de $140 - 100 = 40dB$.

Atténuation globale d'un bruit en courant à la fréquence limite de repliement (dans le cas de l'ADC Arduino Uno)

Nous faisons fonctionner l'ADC à la plus grande fréquence utilisable sans perte de qualité (dans le cas de l'Arduino Uno, $200kHz/13 = 15.4$ kHz). La limite de repliement ou fréquence de Nyquist, correspond à la moitié de la fréquence d'échantillonnage. La fréquence d'échantillonnage est donc ici égale à 15 kHz, donc la fréquence de Nyquist vaut 7,5 kHz. En se référant à la figure 4, nous observons un gain de 33dB à $f_{Nyquist} = 7,5$ kHz. Nous en déduisons une atténuation de $140 - 33 = 100$ dB.

Expression analytique de la résistance du capteur en fonction des éléments du circuit transimpédance

$$R_{capteur} = \left(1 + \frac{R_3}{R_2}\right) * R_1 * \frac{V_{CC}}{V_{ADC}} - R_1 - R_5$$

$$R_5 = 10 \text{ k}\Omega$$

$$R_1 = 100 \text{ k}\Omega$$

$$R_2 = 1 \text{ k}\Omega$$

$$R_3 = 100 \text{ k}\Omega$$

Etude des filtres du circuit

1. Filtre R_1, R_5, C_1

Ce filtre permet de réduire l'impact des Hautes Fréquences sur le circuit.

En négligeant R_5 devant R_1 , nous pouvons calculer la fréquence de coupure théorique de ce filtre :

$$f_{c_{théorique}} = \frac{1}{2\pi * R_1 * C_1} = 16 \text{ Hz}$$

Expérimentalement, nous simulons la réponse du capteur à un balayage fréquentiel, et nous observons le gain en sortie de ce filtre (sonde sur V_{ep}).

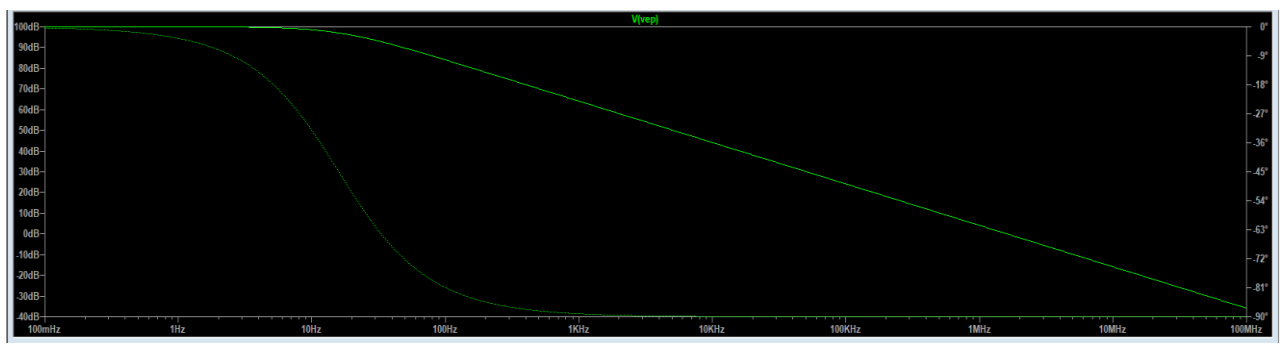


Figure 6. Gain du filtre R_1, R_5, C_1

La fréquence de coupure expérimentale correspond à la fréquence pour laquelle le gain vaut $\max - 3\text{dB}$, c'est-à-dire, $100 - 3 = 97\text{dB}$. Nous obtenons une fréquence de coupure expérimentale

$$f_{c_{exp}} = 15,8 \text{ Hz} \sim 16 \text{ Hz}.$$

2. Filtre R_3, C_4

Ce filtre a pour objectif de réduire le bruit à 50 Hz. Sa fréquence de coupure théorique vaut

$$f_{c_{théorique}} = \frac{1}{2\pi * R_3 * C_4} = 1,6 \text{ Hz}.$$

Expérimentalement, nous simulons la réponse du capteur à un balayage fréquentiel et nous observons le gain en sortie de ce filtre (sonde sur V_{ADC}).

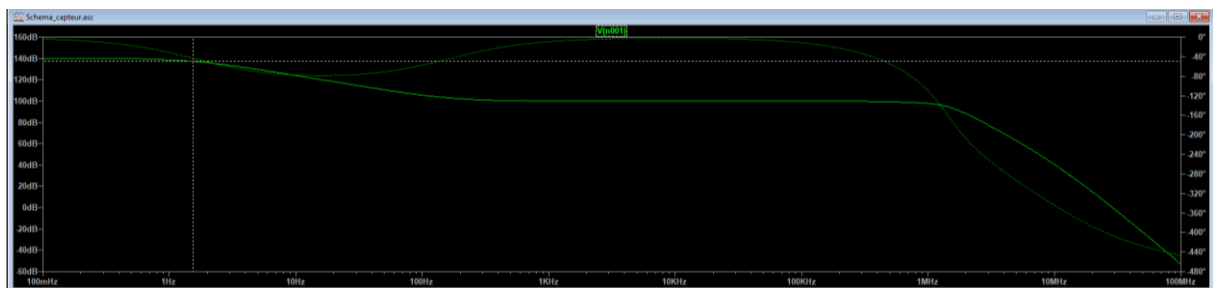


Figure 7. Gain du filtre R_3, C_4 lors d'un balayage fréquentiel

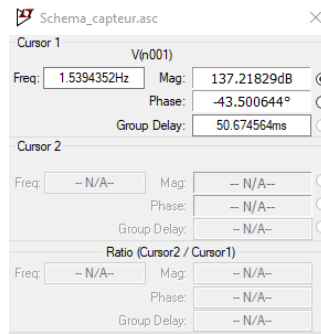


Figure 8. Fréquence de coupure du filtre R3, C4

Expérimentalement, nous obtenons une fréquence de coupure pour ce filtre $f_{c_{exp}} = 1,54 \text{ Hz} \sim 1,6 \text{ Hz}$.

3. Filtre R₆, C₂

Cet étage de filtre permet de « lisser » le signal de sortie. En effet, il retire le bruit introduit en cours de traitement (bruit d'alimentation, bruit d'horloge). Sa fréquence de coupure théorique vaut $f_{c_{théorique}} = \frac{1}{2\pi * R_6 * C_2} = 1,6 \text{ kHz}$. Expérimentalement, nous simulons la réponse du capteur à un balayage fréquentiel et nous observons le gain en sortie de ce filtre (sonde sur V_{ADC}).

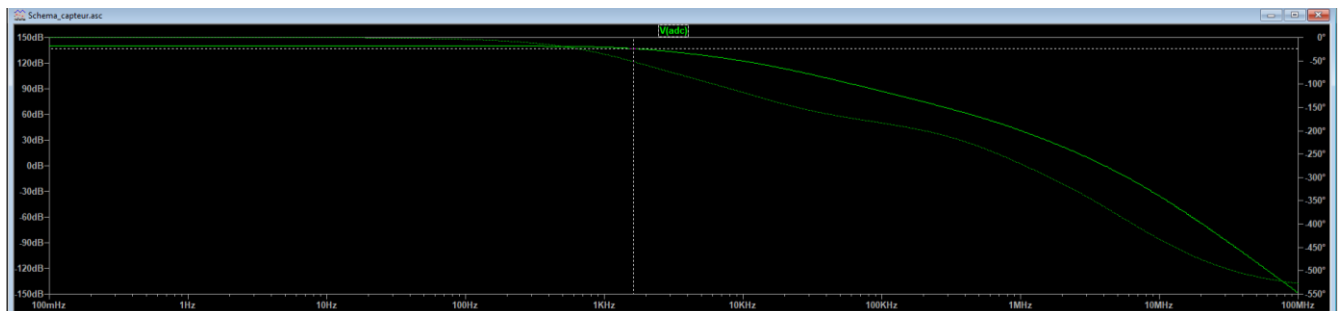


Figure 9. Gain du filtre R6, C2 lors d'un balayage fréquentiel

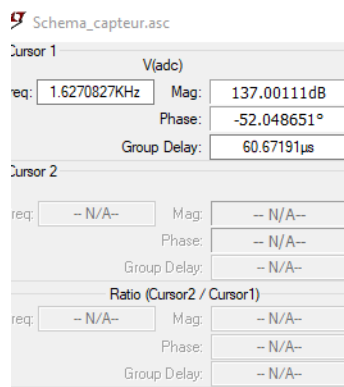


Figure 10. Fréquence de coupure du filtre R6, C2

Expérimentalement, nous obtenons une fréquence de coupure pour ce filtre $f_{c_{exp}} = 1,63 \text{ kHz} \sim 1,6 \text{ Hz}$.

Simulation du potentiomètre digital sur le signal de sortie du circuit

La résistance R_2 du circuit transimpédance est fixe ici. Cependant, dans le circuit réel, cette résistance est remplacée par un potentiomètre digital, c'est-à-dire une résistance variable. Le potentiomètre digital permet d'obtenir une valeur de tension de sortie ADC toujours mesurable par la carte Arduino dont l'intervalle mesurable est entre 0 et 4,8V. Nous avons ci-dessous mesuré la tension ADC de sortie en réponse à un pulse d'amplitude 0,50 à 1V simulant le capteur de déformation :

* $R_2 = 100 \Omega$

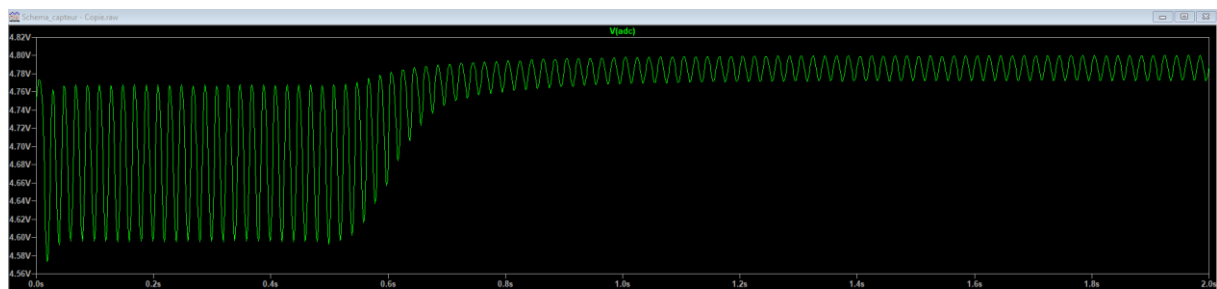


Figure 11. Tension de sortie ADC avec $R_2 = 100 \Omega$

La résistance R_2 étant trop faible, la tension de sortie est très proche de 4,8V donc saturé.

* $R_2 = 250 \Omega$

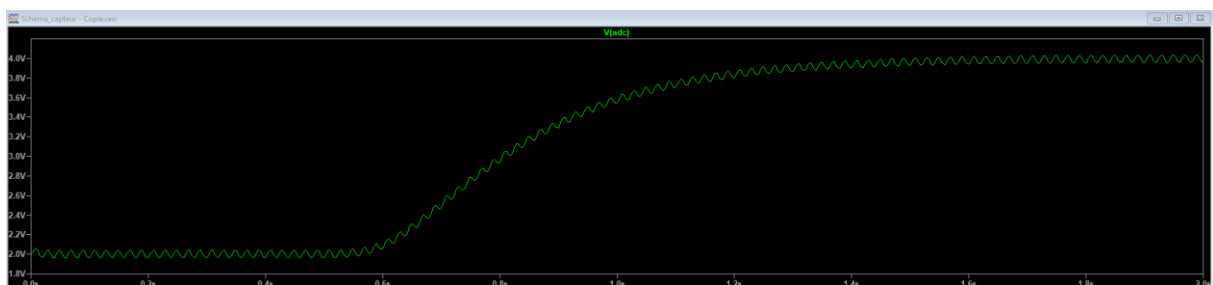


Figure 12. Tension de sortie ADC avec $R_2 = 250 \Omega$

Labrousse Mattéo
Dabadie Raoul

La tension de sortie est d'environ 4V, un risque de saturation peut apparaître lors de la déformation du capteur, il faut donc encore augmenter R_2 pour obtenir une tension de sortie du circuit acceptable.

* $R_2 = 500\ \Omega$

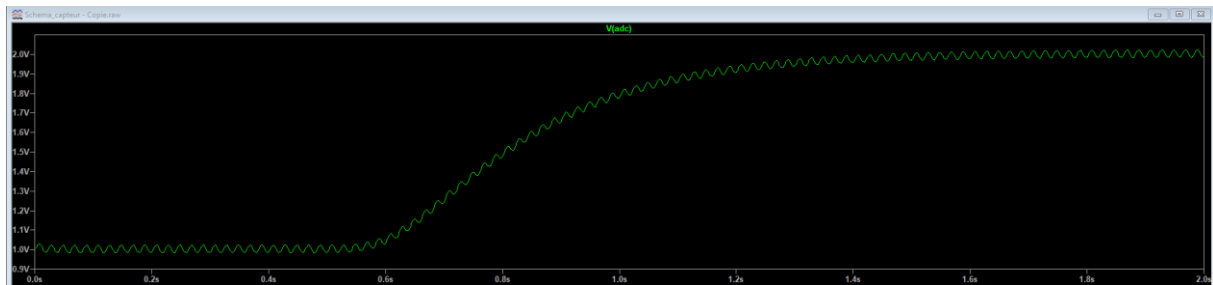


Figure 13. Tension de sortie ADC avec $R_2 = 500\ \Omega$

La tension de sortie est aux alentours de 2V, soit environ la moitié de 4,8V donc nous avons diminuer au minimum le risque de saturation lors de la déformation du capteur.

* $R_2 = 1\ \text{k}\Omega$

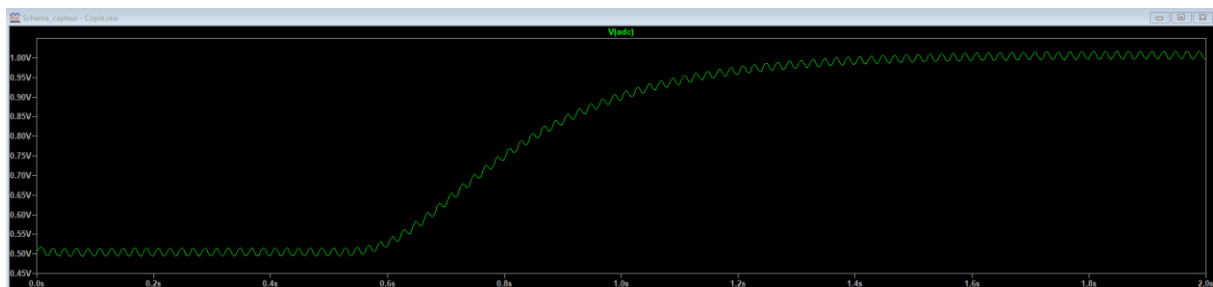


Figure 14. Tension de sortie ADC avec $R_2 = 1\ \text{k}\Omega$

La tension de sortie est de 1V, cette tension est proche de la limite faible mesurable par l'Arduino Uno.

* $R_2 = 2,5\ \text{k}\Omega$

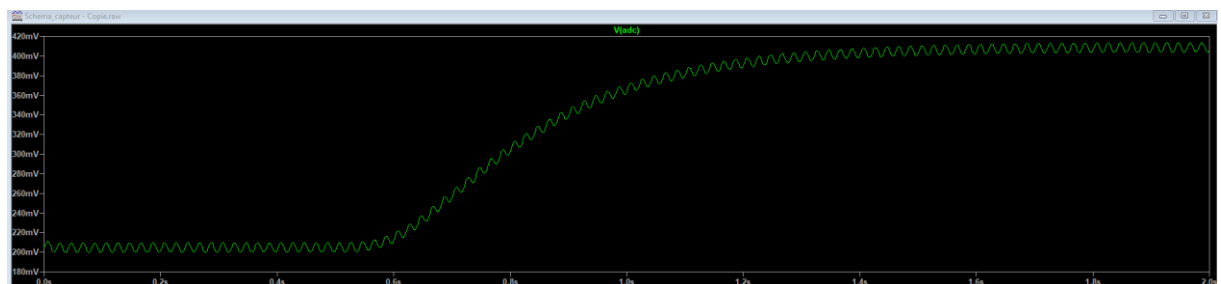


Figure 15. Tension de sortie ADC avec $R_2 = 2,5\text{ k}\Omega$

La tension de sortie est trop faible, la précision de la mesure risque d'être impactée. Il faut une résistance plus faible afin d'acquérir un signal de sortie ADC plus important.

Dans le cas de cette simulation, le potentiomètre digital permettra de choisir une résistance R_2 d'environ $500\ \Omega$ afin d'acquérir une tension de sortie du circuit d'environ 2V et d'être en mesure de réaliser correctement les mesures en minimisant les risques de saturation de la carte Arduino Uno.