"They See Me Rolling" [1]: Wheel Slippage on an Uphill Slope

Ellie McCarthy Florence Townend

Abstract—**Changes in the environment in which a robot operates can drastically affect the performance of many aspects of its function. In this report, we explore the effect of a sloped environment on the "wheel slippage" of a robot, and use a regression model utilising the maximum variation from a control (0 cm slope) case to predict the slope angle a robot travels up in real-time. The experiment highlighted that increasing slope angle makes the behaviour of the robot more unpredictable between trials at each angle, which limits the accuracy of the slope prediction.**

I. Introduction

A robot experiences wheel slippage when the wheel rotation does not translate to the expected translational movement across a surface. Uphill wheel slippage is commonly observed on a slope, when gravity is acting to pull the vehicle against the intended direction of motion induced by the wheel movement. The amount of wheel slippage is a function of variables such as motor power, wheel and surface material, and slope angle. This phenomenon can occur in both downhill and uphill cases, regardless of whether the robot's motors are supplied with power. When no power is supplied to the motors, downhill slippage occurs when the weight of the robot exceeds the frictional force between the wheels and the surface, causing it to slip down the slope.

When the wheels are supplied with power, wheel slippage acts against uphill movement. The wheel slippage can cause a gradual decrease in translational speed, or a sudden decrease in speed as the wheels cease rotation. This is explored by Chen at al. [2] in the design of an aircraft anti-skid braking system to obtain optimal braking performance by preventing wheel locking, minimising tire wear, and reducing wheel slippage. By utilising the longitudinal wheel slippage in a PID (Proportional, Integral, Derivative) control scheme, the presented control algorithm successfully stabilises the wheel slippage around a given equilibrium point.

Wheel slippage does not only affect the movement and efficiency of a robot, but also affects how accurately the robot can measure how far it has travelled. This measurement is reliant on the how much the wheels have turned; hence if the wheels slip and the robot moves without wheel rotation, the distance measurement will suffer as a consequence. A robot may measure the distance it has travelled for a number of reasons: perhaps it is measuring a distance to relay back to the user, or it might be programmed to travel a certain distance and carry out a task when that distance is reached. An investigation into how the surface gradient affects the extent of wheel slippage could inform on how best to prevent wheel slippage from occurring. The experiments detailed in this report were performed on a Pololu Romi 32U4, and the robot used will be referred to as the "Romi".

A. Hypothesis Statement

We hypothesise that by increasing the angle of the slope that the Romi is ascending, the Romi will experience wheel slippage, making the relationship between input power and movement less deterministic than if it were travelling on a flat surface. Machine learning techniques, such as regression, could harness this hypothesised relationship between slope angle and wheel slippage to predict the angle of the slope being traversed. A secondary hypothesis in this report is that, if the Romi is travelling in a straight line, the accuracy the distance measurements will decrease as slope angle is increased.

II. Implementation

In order to predict the angle of the slope the Romi is ascending, we must first investigate the behaviour of the Romi on slopes of varying steepness through experimentation. We must add controls to our experiment to make the findings valid.

A. Rotary Encoders

The distance travelled by the Romi can be measured using readings from two rotary encoders. These rotary encoders tell the user if the wheel is rotating and in which direction it is rotating (clockwise or anticlockwise). We have constructed an Interrupt Service Routine (ISR) to track the states of each encoder. The ISR momentarily pauses the main-loop code running in the central processing unit and implements a quick encoder script to monitor the encoder pulses. Polling the encoder counts using a short interval was also a viable option, but was disregarded due to the possibility of missing vital encoder counts.

The main-loop code utilising the Romi's quadrature encoder evaluates the change in encoder counts at 40 ms intervals and calculates an estimated speed for the right wheel. The speed of the right wheel is used as an input for the PID controller for the left wheel to ensure straight line movement, further discussed in Section II-B. Furthermore, as each encoder count represents approximately 0.015 m in forward travel for the Pololu Romi 32U4, the distance travelled can be measured by the change in encoder counts between the start and stop of measurement, multiplied by 0.015. By comparing the expected distance travelled and the actual distance travelled, we can determine whether any wheel slippage occurs when the Romi moves up a slope.

The change in encoder count is our experimental dependent variable, so it is important to consider any possible sources of error affecting encoder count values during our experiments. As the encoder count data we collect is obtained over multiple trials, repeatability is an important concept. Unlike capacitive encoders which accumulate error over time, our magnetic-based encoder has been designed to ensure they are not subject to cumulative error. Furthermore, the quadrature output of the encoders used in our Romi have a higher resolution than typical incremental encoders, meaning we can be confident in the resulting resolution of our collected data. Mechanical imperfection will also be a likely, and unavoidable, source of error. Undertaking multiple trials of each experiment, identifying anomalies, and calculating averages are important steps to be taken in mitigation.

B. PID Control

As the input wheel power and the resulting wheel speed are approximately proportional, one would expect that if the left and right Romi motors were provided the same power, then the Romi should travel in a straight line without turning. However, discrepancies between the hardware of each motor mean that a single power value will produce different outputs in different motors, and using the same power input for two motors can not be relied upon to produce straight line motion.

For this report's experiment, the Romi needs to travel in a straight line because "start" and "stop" lines for the measurement are perpendicular to the direction of motion. If the Romi were to rotate while it travels between those lines, then the Romi may travel further than the specified controlled distance before detecting the "stop" line.

A PID controller is commonly used to regulate the power given to each motor in order to produce straight line motion. A PID controller is a closed-loop approach, which means that the controller automatically regulates the power being given to motors by responding to outputs of the system. The inputs needed for the controller to function are the demand, which is the desired encoder count per second that we want the wheel to rotate at, and the measurement, which is the wheel's current encoder count frequency.

In this experiment, it is not possible to use a traditional PID controller to ensure the Romi moves in a straight line because we need to record the encoder counts for our investigation. Therefore, we cannot specify how many encoder counts per second we want our wheel to rotate at because that would jeopardise the validity of our results.

In order to avoid this problem, we will be using a PID controller for just one of the wheels, where the demand encoder count frequency will be the encoder count frequency measurement from the other wheel. By doing this, we are able to measure the encoder counts from the independent wheel for our investigations without interfering, but we are also able to ensure that the wheels are speed-matched and the Romi travels in a straight line. In our experiments, we have chosen the independent wheel to be the right wheel, and the left wheel is matching the right wheel's speed. This is an arbitrary choice and, to substantiate this claim, we conducted trials for the left wheel being independent and we found no identifiable difference.

III. Experiment Methodology

As an overview of our experiment, the Romi travels a fixed distance along a surface and a stream of right-wheel encoder counts is recorded. In order to investigate how the slope of the surface affects wheel slippage, the surface is tilted so that the Romi travels uphill along the same fixed distance but at different gradients, as illustrated in Figure 1.

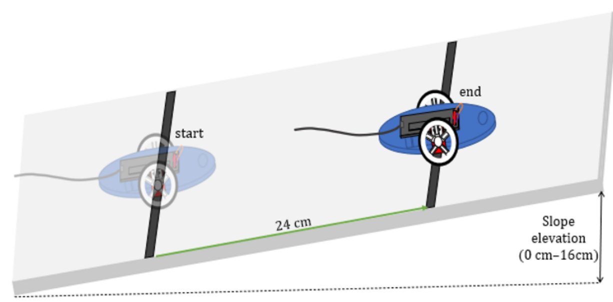

Fig. 1. A side view of the experiment set up.

One iteration of the experiment is conducted in stages:

1) The Romi travels forwards until it reaches a black line on the surface, detected using the QTR-3A Reflectance Sensor Array.
2) Once the Romi has travelled across this black line and detects the white surface again, data collection is initiated and the Romi begins to travel the fixed distance (24 cm).
3) The Romi travels in a straight line and outputs the encoder counts at small regular 40 ms intervals.
4) When the Romi detects the beginning of the second black line, it stops outputting the encoder count measurements and comes to a stop.
5) The data gathered via USB cable is stored and the Romi reset manually to the starting position below the first black line.

A. Controlled Variables

The validity of the findings in this report is ensured by controlling the variables that could have an effect on the results. Specifying the measures to control these variables also allows this experiment to be reproduced accurately.

1. Hardware

- **Batteries**: Power level in the Romi can affect its performance. The batteries were gradually replaced two at a time throughout the experiment in an attempt to keep the Romi wheel power approximately constant.
- **Components**: The Romi's QTR-3A Reflectance Sensor Array [3] was used for detecting the "start" and "stop" lines throughout the experiment. This array consists of 3 separate sensors, each containing a phototransistor and an infra-red LED, which return reflective values in the range of 0 to 1023.
- **Robot**: Pololu Romi 32U4.

2. Software

- **PID**: The PID method outlined in Section II-B requires gain values that are tuned to the task. Our gain values are as follows: $K_P = 50$, $K_I = 5$, $K_D = 4$. These values were tuned by eye when observing the Romi's movement over a flat surface and plotting left wheel speed against right wheel speed.
- **Encoder count output**: at a constant 40 ms interval.
- **Black "start" / "end" line detection**: The line sensor component requires a threshold sensor value to distinguish between black and white surface colours. A reflective value of 200 was chosen as the threshold through observation of the line sensor output readings when travelling between black and white surfaces. To minimise the false detection of black lines due to extraneous factors such as surface irregularities, each of the three sensors were required to meet this threshold separately.

3. Environment

- **Surface**: The surface on which the Romi travels stays constant throughout the experiment and was chosen for its smoothness, where wheel slippage would be more prevalent and interesting to investigate.
- **Distance travelled**: In each run of the experiment procedure, the Romi travels 24 cm between two black lines marked with electrical tape, the placements of which do not change. This set up is shown in Figure 1.

4. Task

- **Data collection**: A stream of encoder counts is required to allow to Romi to predict the slope angle as it travels uphill. This data is collected from the right wheel of the Romi at 40 ms intervals via the USB cable. During the experiment, the USB cable is elevated to avoid contact with the slope's surface, whilst maintaining slack to ensure little net force is applied to disrupt the natural motion of the Romi. This is not ideal because the weight of the cable might inhibit the motion. However, to record a continuous stream of encoder counts, this is the simplest way. As this USB-based data collection method is used for every trial, the systematic error should be constant and not affect the relative findings between runs. The streamed data collected using this method is used as training data for the prediction model.
- **Number of trials**: A total of 10 trials are conducted at each height increment. This number is the result of a trade-off between data volume, experiment duration, and battery usage.
- **Chosen experimental increment**: A preliminary investigation highlighted a significant variation in final encoder count for a fixed slope height, so increasing the height in 2 cm intervals was used to maintain resolution whilst allowing a clear increase in final encoder count to be observed across heights. These vertical heights explored were: 0 cm, 2 cm, 4 cm, 6 cm, 8 cm, 10 cm, 12 cm, 14 cm, and 16 cm. The maximum slope elevation was chosen to be 16 cm as the Romi was unable to ascend any slopes steeper than this without increasing the input power.
- **Direction of travel**: Each trial, the Romi starts perpendicular to the start line and ascends a slope. Uphill motion has been chosen because the rear ball caster system provides an extra mode of constant contact with the surface, ensuring stable motion. During brief experimentation with downhill motion, we discovered that the resulting loss of contact between the ball caster and the surface caused the Romi to oscillate about the wheel axis, making the front end of the Romi bounce against the surface during motion. The intention is that the Romi travels perpendicular to the start and stop black lines, but the angle of travel is subject to human error when the Romi is placed by hand onto the test surface. By taking multiple trials at each height, we hope to limit the error introduced during Romi placement.
- **Input wheel power**: The Romi's right wheel digital-to-analog converter is provided with a value of 33, corresponding to approximately 0.9 V, whereas the power input to the left wheel is determined by the PID controller.

B. Independent Variables

In this experiment, although we investigate how the angle of the slope affects wheel slippage, we have chosen the vertical height of the end of the test surface to be the independent variable. We have made this decision because a small change in slope angle corresponds to a large change in vertical height, and this larger increment size for the vertical height means that the measurement is less affected by human uncertainty.

C. Dependent variables

The dependent variable in this experiment is the encoder counts at regular time intervals across the fixed distance travelled. We have chosen to use a sequence of encoder counts throughout the experiment to allow visualisations of wheel slippage by plotting encoder counts against time.

IV. Results

A. Encoder Counts

The results of our experiments are displayed in Figure 2. Each encoder count sequence is illustrated by a different coloured line, where each set of 10 trials for a given height share the same colour, as indicated by the key.

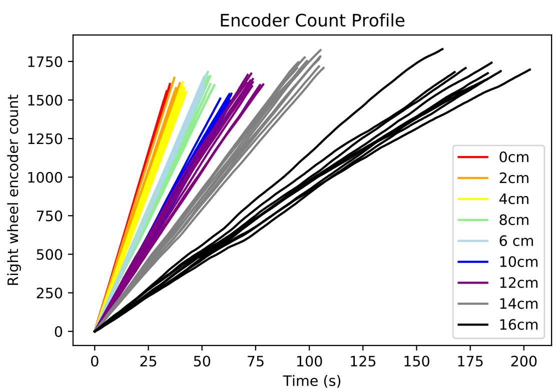

Fig. 2. All the collected encoder counts from 9 elevations at a constant power of 0.9 V travelling a fixed distance. The lines show the encoder counts at each time for each elevation trial.

The clearest trend from this graph is the decrease in speed with increasing slope angle, indicated by the decrease of encoder count gradient throughout the 0 cm – 16 cm trials. This decrease in Romi speed occurs because motor power is one of our control variables, and our PID controller acts only to encourage straight-line movement. This trend does not appear to be linear, as the Romi undergoes a greater decrease in wheel speed between the 14 cm and 16 cm trials, than any other 2 cm height interval.

Another, more subtle, trend is the increase in wheel speed variability with increasing slope elevation. The wheel speed variation for a given set of 10 trials is indicated by the area spanned by the 10 encoder count trajectories. We can see from Figure 2 that the encoder count trajectories of the 12 cm, 14 cm, and 16 cm trials span a wider area than the trials conducted on the shallower slopes. This finding shows the movement of the Romi becomes more unpredictable when it ascends steeper slopes.

We were unable to observe any significant wheel slippage in any of the trials with elevations in the 0 cm – 12 cm range, shown by each of the 0 cm – 12 cm trials recording a final encoder count between 1500–1750. However, increasing the vertical elevation to 14 cm causes wheel slippage to occur, shown by the grey 14 cm final encoder counts being greater than 1750.

Wheel slippage was also observed throughout the 16 cm set of trials, but does not noticeably increase from the 14 cm set of trials, with both sets of encoder count trajectories ending in the same 1750+ range. However, the wheel speeds observed in the 16 cm set of trials show a significantly higher variability, indicating that, unlike previous trials, the Romi speed does not stay constant within each trial. When conducting the experiment at the 16 cm slope elevation, we observed a new type of Romi behaviour. In addition to the gradual wheel slippage seen previously at the 14 cm set of trials, when traversing the slope with a 16 cm elevation, the Romi would repeatedly pause before continuing its journey. Not only does this explain the increased Romi speed variability, it also explains why the Romi takes over twice as long to complete the fixed distance at 16 cm elevation, compared to 14 cm elevation.

The next step in this report is to find an aspect of the Romi's uphill travel that can accurately predict the angle of the slope. We will use the information and inferences found in Figure 2 to develop a prediction method.

B. Slope Prediction

We want to be able to predict the angle of the slope that the Romi is ascending. It is important to make real-time predictions because it will allow the Romi to use the data it gathers to change its behaviour, such as reducing acceleration on steeper surfaces to avoid wheel slippage. The feature of the Romi's uphill travel that we will use to predict the angle of the slope is the variation within the encoder count stream. This variation is calculated by joining the first and last data points in an encoder count stream to make a straight line (acting as our perfect encoder count stream with no wheel slippage), and calculating the difference between that straight line and the measured encoder counts. An illustration of the variation is shown in Figure 3, where the true encoder count stream (red) is compared to the straight-line version (black). The left figure shows this for flat travel and the right shows this for a vertical elevation of 16 cm. We can see that the steeper slope causes the encoder counts to vary from the straight line, indicating wheel slippage has occurred.

Two ways of quantifying this variation have been explored in this report: mean error and maximum error. Variations above and below the straight line have been treated the same and therefore the absolute variation from the straight line has been considered.

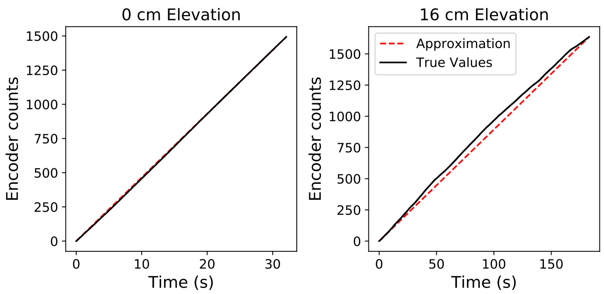

Fig. 3. Illustration of slope prediction metric. (Left) Encoder counts vs time for the full journey (black) and the straight line approximation (red) for a flat surface. (Right) Full journey encoder counts and straight line approximation for 16 cm elevation.

1) Whole Journey Prediction: We assessed the validity of the variation measure by attempting to the predict the vertical elevation of one side of the surface after a complete journey across the fixed distance (experiment setup shown in Figure 1). Splitting the data into train and test sets with a two-thirds and one-third ratio, we found a regression equation to fit the train data. We attempted to avoid overfitting by visually inspecting the regression against the train data, and making sure that it was not trying to capture every data point unnecessarily. The noisiness of the data due to human error also helps avoid overfitting [4].

The regression equation used for all the regression tasks in this report is of the form

$$\bar{y} = f(x) = ax^{\frac{1}{2}} + bx + c, \tag{1}$$

where y is the predicted vertical height (analogous to slope angle), x is the variation measure, and a, b, c are parameters tuned to optimise the fit of the regression to the train data.

Mean-Squared Error (MSE) and Mean-Absolute Error (MAE) are used to assess the model performance. MSE and MAE are defined as $\sum_{i=1}^{N}(\bar{y}_i - y_i)^2$ and $\sum_{i=1}^{N}|\bar{y}_i - y_i|$ respectively, where $\bar{y}$ is the predicted vertical height, y is the true vertical height, and N is the number of test data points.

Figure 4 (left) shows the results from our regression when calculating straight-line variation as the mean absolute distance from the straight line, and Figure 4 (right) shows the results when using the maximum absolute distance. This figure shows the test data in green, and the predicted vertical heights are shown in red. Table I shows the results from both of these methods. Using the maximum of the distances from the straight line produced a better fit for the vertical heights, so we will be using maximum distance as our method for calculating straight-line variation in our next section.

2) Simultaneous Prediction: The next step in this report is to make the slope predictions in real time and not be reliant on the finished set of encoder counts. This will make the prediction more applicable to scenarios other than a simple start-stop course. A prediction is made once the encoder count value reaches 300 more than when the last prediction was made (or when the Romi started moving). This interval of encoder counts may take place over a longer time, but because we are not summing the variations, just taking the maximum, this should not affect the prediction's credibility. Figure 5 shows the results for continuous vertical height prediction, when using the maximum absolute distance from the straight line as the predictive variable. We are using the maximum distance because it performed better in the previous trials in Table I. The MSE for this experiment is 14.79 and the MAE is 3.07 cm, which shows that real-time prediction has improved performance over using the whole journey of the Romi.

Method	MAE (cm)	MSE
Maximum	3.39	19.13
Mean	3.61	19.42

TABLE I
WHOLE JOURNEY PREDICTION RESULTS.

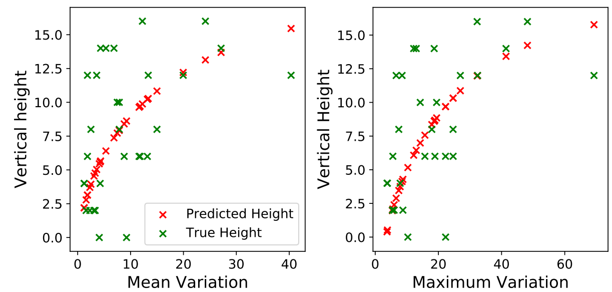

Fig. 4. The true vertical elevations (cm) (green) compared to the predicted elevations (red) when using mean straight-line variation (left) vs maximum straight-line variation (right) as the independent predictive variable.

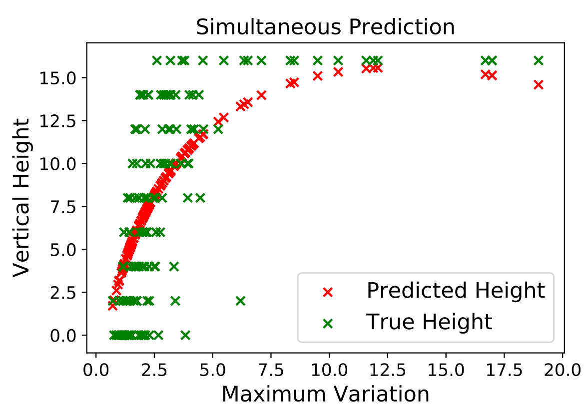

Fig. 5. Results from the simultaneous vertical elevation (cm) prediction when using maximum straight-line variation as the predictive variable.

V. Discussion and Conclusion

A. Experiment Evaluation

Our experiment could be improved by increasing the number of trials conducted at each height. This would provide greater confidence in the consistency of our results and ensure the key trends described in Section IV-A were not due to random events. However, repeating the number of trials would require a closer monitoring of the Romi's battery power, as this accelerated battery drainage would introduce further systematic error to our investigation.

By implementing the left wheel PID controller, we hoped to mitigate the operational discrepancies between the motors, but even after tuning the PID coefficients on the non-elevated test surface, we were unable to fully eliminate the motor imbalance, meaning the Romi often failed to intersect the "stop" line at 90 degrees. Consequently, the Romi may have travelled further than 24 cm at each trial, meaning an elevated encoder count between heights may not have only been due to the occurrence of wheel slippage.

B. Slope Prediction Evaluation

We found an exciting result: we can predict the vertical elevation of the surface of the Romi's uphill travel in real time with a mean absolute error of 3 cm. A benefit of the method is that the straight-line variation of encoder counts against time does not take the speed of the Romi into account because we are not interested in the gradient of the line or how long it takes to increase encoder counts, only how variable the profile of encoder counts against time is.

However, limitations of this regression method stem from only exploring one motor power in the data collection experiments. It would be interesting to explore whether new predictive models would need to be created for each motor power or speed used in a Romi's journey in order to predict the slope. Additionally, the method by which we are creating the straight-line version of the encoder count sequence (joining the first and last data points with a straight line) could be too simple; perhaps a more sophisticated linear best-fit line through the data would produce a better approximation to the straight-line variation. The data for the slope prediction in this report is taken from a full journey of a Romi from start to the stop line, as seen in Figure 1, with the USB cable attached for data collection. However, since this slope prediction can work simultaneously with the Romi travelling, this USB cable would not need to be attached in future applications and we could explore how that affects the predictions themselves.

Exploration on how this slope prediction generalises to additional motor powers and environments, such as diagonal movement up a slope or different surfaces, is key to understanding how best to apply this prediction in order to improve a robot's function. Implementing a PID control system that, when the slope angle has been predicted, alters the speed of the Romi in order to specifically avoid wheel slippage is the best application of this slope prediction. This PID system would improve the measurement of distance travelled by the Romi because the error in the measurement due to wheel slippage would be mitigated. In order to make the slope prediction applicable across many environments and tasks, it would be useful to explore how the prediction is affected by a closed PID speed control system, rather than our one-wheel PID implemented in this report.

C. Conclusion

In this report, we have conducted experiments on how behaviour of a robot when travelling uphill is influenced by the angle of the slope. We found that wheel slippage occurred at increased slope angles, which momentarily caused the robot's wheels to turn but no distance to be covered. The vertical elevation of the slope can be predicted from the fluctuating relationship between cumulative encoder counts and time, with a mean absolute error of approximately 3 cm. This prediction method can be implemented during the robot's travel and updated multiple times during a journey, allowing the robot to use this prediction to adjust its behaviour using the information it learns about its environment. Future applications could include creating a wheel slippage PID that uses the slope prediction to avoid wheel slippage in order to improve accuracy of distance measurement by the robot using encoder counts.

References

[1] Chamillionaire, "Ridin." https://www.allmusic.com/album/ridin-mw0001693793, 2006.

[2] M. Q. Chen, W. S. Liu, Y. Z. Ma, J. Wang, F. R. Xu, and Y. J. Wang, "Mixed slip-deceleration pid control of aircraft wheel braking system," *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 160–165, 2018. 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID 2018.

[3] Pololu Corporation, "Pololu QTR Reflectance Sensors." https://www.pololu.com/product/4243, 2001–2021.

[4] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.