Discovering empirical equations from robot-collected data

Kuang-Ming Huang & Jan M. Żytkow[†]

[†] Dept. of Computer Science, University of North Carolina, Charlotte, NC 28223
and Institute of Computer Science, Polish Academy of Sciences
zytkow@uncc.edu

Abstract. Discovery of multidimensional empirical equations has been a task of systems such as BACON and FAHRENHEIT. When confronted with data collected in a robotic experiment, BACON-like generalization mechanism of FAHRENHEIT reached an impasse because it found many acceptable equations for some datasets while none for others. We describe an improved generalization mechanism that handles both problems. We apply that mechanism to a robot arm experiment similar to Galileo's experiments with the inclined plane. The system collected data, determined empirical error and eventually found empirical equations acceptable within error. By confronting empirical equations developed by FAHRENHEIT with theoretical models based on classical mechanics, we have shown that empirical equations provide superior fit to data. Systematic deviations between data and a theoretical model hint at processes not captured by the model but accounted for in empirical equations.

1 Robotic experiment and challenges of real data

We describe a discovery mechanism that makes several improvements over the BACON-like search for multidimensional empirical equations. It has been motivated by an impasse reached by the existing systems, such as BACON and FAHRENHEIT, applied to data produced automatically in robotic experiments.

1.1 Challenges to BACON-like search for equations

BACON-like search for multidimensional equations proceeds step by step, adding one independent variable at a time. Suppose the experimenter collects data by setting the values of three variables x_1, x_2, and x_3, and measuring y. The final goal is an equation that combines y, x_1, x_2, x_3. It can often be expressed in a form convenient for predictions: $y = f(x_1, x_2, x_3)$. At any step of BACON's generalization from data to equations, independent variables can be divided in three categories: (1) those that have already been used in an equation (for example, x_1), (2) one variable that is being added (for example, x_2), and (3) those variables which have been kept constant in all experiments (x_3).

Generalization to x_2 is triggered when an equation has been found for y and x_1. The generalization process starts from data collection: for each value of x_2

an equation is sought for y and x_1. Data collection is successful when BACON reaches an externally selected number n of equations of the same algebraic form $f(y, x_1, A_1, \ldots A_k) = 0$, which may have different values of parameters $A_1, \ldots A_k$. Each equation corresponds to one value of x_2. The following table summarizes the "higher level" data used for generalization. Each a_{ij} represents a j-th value of parameter A_i:

x_2	A_1	$\ldots$	A_k
v_1	a_{11}	$\ldots$	a_{k1}
v_2	a_{12}	$\ldots$	a_{k2}
$\ldots$	$\ldots$	$\ldots$	$\ldots$
$\ldots$	$\ldots$	$\ldots$	$\ldots$
v_n	a_{1n}	$\ldots$	a_{kn}

Now the task is to find k equations of the form $A_i = g_i(x_2)$, $i = 1, \ldots, k$ that link x_2 with each of $A_1, \ldots A_k$, one at a time. Those equations can be used to eliminate $A_1, \ldots A_k$ in the original equation $y = f(x_1, A_1, \ldots A_k)$. As a result, the equation for y uses two independent variables x_1 and x_2 plus several new parameters $B_1, \ldots B_m$, which can be used to generalize the equation to the next independent variable.

BACON works fine when it discovers a single equation per dataset, the same for all data. Langley et al. (1987) offer many examples of successful search, but pay little attention to the many ways in which the BACON search may fail. Even a simple failure at one of many steps causes the whole system to fail and halt. In this paper we describe solutions to the following problems:

1. Search results in multiple alternative equations, that offer acceptable fit to data. If equations of different form are best for different datasets, which equation should be selected for generalization?

2. No equation of common form is acceptable for each dataset. This prevents a meaningful generalization, as all values of any given parameter A_i such as the slope of linear equation, must have the same meaning, so the induction over different values makes sense.

3. Equations can be evaluated when the measurement error is known. In BACON, the value of error is provided from the outside, but it should be determined by experiments since it is specific to a given experiment setup. Evaluation may be overly demanding or too permissive if the system uses wrong values of error. A discovery method should use data to infer the error and then propagate the error so that it applies at all stages of the discovery process.

1.2 Experiment setup

Let us distinguish between two meanings of experiment. In the first meaning, an experiment includes the investigated empirical system S, the manipulating and measuring equipment and the strategy of using them. We shall call it a setup experiment. In the second meaning, an experiment is a single cycle of interaction between the experimenter and empirical system S. The cycle consists of creating a particular state of S, determined by the values of some variables, and in measuring the response of S in terms of other variables. In this paper we consider a discovery system that performs many experiments in the second sense, by varying objects and their properties within the fixed setup.

1.3 Empirical space

BACON and FAHRENHEIT operate within a fixed empirical setup. The scope of their experiments can be represented by a simple formal space. Consider M control variables $x^1, \cdots, x^M$ and N dependent variables $y^1, \cdots, y^N$. Each $x^i, i = 1, ..., M$, is limited to a set of values X^i, that is to the scope of manipulations. Each $y^i, i = 1, ..., N$ can be measured within a set of values Y^i. The values of all variables form a Cartesian product $\mathcal{E}$ of $M + N$ dimensions. Each dimension is a segment of values between a minimum and a maximum.

Experiments are the only way for obtaining information about $\mathcal{E}$. Each experiment consists of enforcing a value for each independent variable $x^i, i = 1, \cdots, M$, and of measuring afterwards the values of $y^j, j = 1, \cdots, N$.

The values of each variable carry empirical error. In this paper we will only consider error of dependent variables. The values of error, ε^i, can vary across Y^i. They can be determined by FAHRENHEIT in the course of experimentation, as we shall see later. Because of the finite size and the minimum grain level determined by the error of each variable, the space $\mathcal{E}$ is finite from the perspective of possible manipulations and measurements.

1.4 Relation to other systems

Although the new methods described in this paper actually expand FAHRENHEIT (Żytkow, 1996), in this paper we only consider a subset of FAHRENHEIT similar to BACON-3 (Langley et al. 1987). In consequence, for the sake of simplicity, we describe the new methods as additions to BACON-3.

Just as BACON-3 uses BACON-1, FAHRENHEIT uses Equation Finder (EF; Zembowicz & Żytkow, 1991) to generate bi-variate equation. Bi-variate equation finding has been the most popular area of automated discovery (e.g. Nordhausen & Langley, 1990; Moulet, 1992; Dzeroski & Todorovski, 1993). EF can be distinguished by its broader space of equations and systematic use of error.

Experiments conducted by FAHRENHEIT and simulated experiments of BACON occur within a given setup. Kekada (Kulkarni & Simon, 1987) offers a broader perspective on experiments, designing them within a number of setups and providing links between experiments in different setups.

Nordhausen and Langley's IDS (1990), although capable of deriving multidimensional equations and placing them in sophisticated theories, does not control experiments and neither deals with the choices among many alternative equations nor consistently handles empirical error.

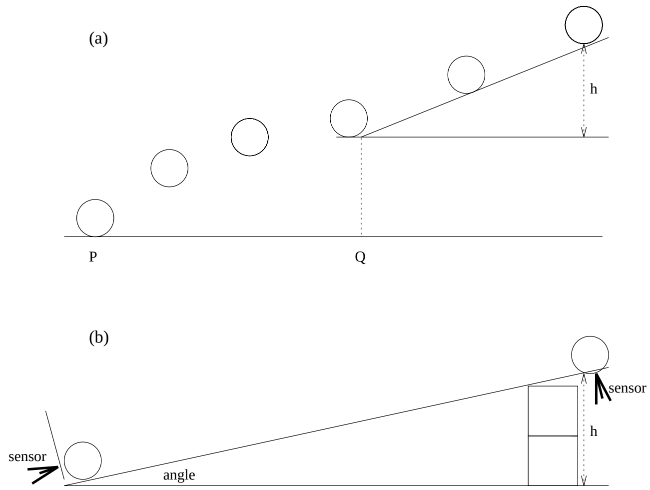

Fig. 1. (a) Galileo's experiment. A ball is rolling down the inclined plane. The experimenter controls height h. When the ball reaches the bottom of the ramp it assumes horizontal direction and falls to the floor at point P. The distance PQ is proportional to velocity. (b) Our experiment. A cylinder moves a fixed distance between a sensor at the top of the ramp and the sensor installed in the buffer at the bottom.

2 Experiment description

In distinction to our previous automated experiments (eg. Zytkow, Zhu & Hussam, 1990) in the domain of chemistry, this time we used a robot arm and the measurement of mechanical motion. We also wanted to evaluate our robotic discovery system on a well-known experiment that influenced the foundation of mechanics.

2.1 Galileo's discovery

Galileo investigated the motion of objects rolling down an inclined plane. A ball, released from the top of the inclined plane depicted in Figure 1a, rolled down and finally hit the bottom. Galileo was not able to measure the accurate time at which the ball reached the bottom, but he could indirectly measure the final velocity. He attached a "jumping board" to the bottom of the inclined plane, so that reaching the bottom the ball assumed horizontal velocity and fell through the air to hit the floor at point P. Galileo used the distance PQ between the jumping board and point P to calculate the velocity of the ball at the bottom of the slope. After a sequence of experiments in which he varied the height h from which the ball started to roll, he derived a theory of velocity v,

$$v = \sqrt{ah}\,, \qquad a = constant. \tag{1}$$

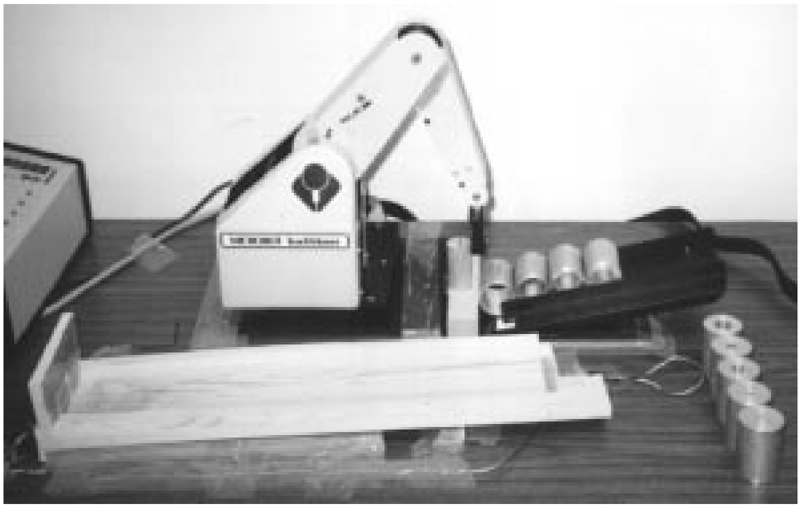

Fig. 2. The experiment setup. The robot arm is reaching for the first cylinder. Four cylinders have been placed in the cylinder container, while the other five can be seen at the lower right corner of the picture. Holes of different diameter are readily visible. The inclined board has been set at the lowest angle. The board has a bumper attached at the lower end on the left to hold the bottom touch sensor and stop the cylinders.

Galileo's theory influenced the foundations of modern mechanics. It was empirically inaccurate, however, partly because he did not consider momentum of inertia of the rolling body.

2.2 Experiment setup

In our experiment we used the inclined plane, but a precise time measurement has been easier for us than the measurement of velocity. Two touch sensors, to signal the beginning and the end of the process, have been placed at fixed points at the top and bottom of the inclined plane at distance D (50.2cm) and attached to the computer (cf. Figure 1b, 2). The equivalent of Galileo's equation (1) expressed in terms of time t, angle α and Earth acceleration g is,

$$t = \sqrt{\frac{2D}{gsin(\alpha)}}. \qquad (2)$$

In our experiments we also wanted to capture the influence of angular momentum on motion. Instead of a ball, we used nine cylinders of the same length (5.9cm) and equal external diameter (4.5cm), with holes of different radius drilled symmetrically through the ax of each cylinder. Their internal diameters differ from 0cm to 4cm by 0.5 cm increments (cf. Figure 2).

Since the fixed locations of touch sensors determined the scope of motion, instead of various locations on the fixed board that Galileo used as his initial states, we varied the angle of the inclined plane. Altogether we used five angles.

Thus, the empirical space consisted of two independent variables (angle and mass) and one dependent variable (time difference between the final and the initial state). The robot arm placed a cylinder of mass m_i at the location of the top sensor (initial state) and then picked it up at the final state, at the location of the bottom sensor.

3 Discovery process

We linked the operational procedures that transport the cylinders and measure time to the FAHRENHEIT discovery mechanism (Żytkow, 1996). For the purpose of this experiment, a subset of FAHRENHEIT has been used, essentially equivalent to BACON-3 (Langley et.al., 1987). Since BACON-3 mechanism is widely known, we will remind it only briefly and concentrate on the new elements.

3.1 Experimentation

The experiments can be summarized as a 3D nested loop:

```
FOR angle FROM angle-1 TO angle-5
   FOR mass FROM mass-1 TO mass-9
      FOR repeat FROM 1 TO 50
         move mass on top of board at angle
         release gripper
         start timer at interrupt from top sensor
         time(angle, mass, repeat) = read timer at interrupt from bottom sensor
```

Nine objects, five angles and fifty repetitions for each pair led to 2250 experiments. At about 30 seconds per experiment, data collection took about 20 hours.

3.2 Error detection

For each cylinder and each angle, the mean value of 50 repetitions has been used as the value of time, while the doubled value of the estimated standard deviation has been used as time measurement error.

3.3 The use of empirical error

The vast majority of discovery systems have been simplistic in handling empirical error. Although many systems include error parameters, they disregard the variety of ways in which error should be used throughout the discovery process.

Consider the data in the form $D = \{(x_i, y_i, e_i) : i = 1, ..., N\}$ and the search for equations of the form $y = f(x)$ that fit (x_i, y_i) within the limits of error e_i. For the same data $\{(x_i, y_i) : i = 1, ..., N\}$, the smaller is the error e_i, the closer fit is required between data and equations. Among the equations that fit the data at error e_i, only some will be acceptable at a smaller error, so that a smaller error may reduce the ambiguity of search for equations.

Knowledge of error, prior to the search for equations, is used in several ways by Equation Finder (EF: Zembowicz & Żytkow, 1991):

1. When the error varies for different data, $\chi^2 = \sum_{i=1}^{N} \left(\frac{y_i - f(x_i, A_1, \ldots, A_q)}{\sigma_i} \right)^2$ (the weighted chi-square value) is used to compute the best values $a_1, \ldots, a_q$ for model f, enforcing better fit to the more precise datapoints. $\sigma = e_i/2$.
2. Error is used in the evaluation of equations. Error of each datum can be interpreted as the standard deviation of the normal distribution of y's for the value of x. For each equation E, knowledge of that distribution permits the computation od the probability that the data have been generated by adding the value of y obtained from E and the value drawn randomly from the normal distribution defined by the error of each datum. This is a plausible null hypothesis for testing, as normal distributions are often a good approximation of error.
3. As many other equation-finding systems, EF generates new terms by transforming the initial variables x and y. Error values are propagated to the new terms and used in search for equations that use those terms.
4. Error is propagated to the values of $A_1, \ldots, A_q$ in the fitted equations.
5. If the error of a parameter is larger than the absolute value of that parameter, EF assumes that the parameter value is zero. If this happens to the parameter at the highest degree polynomial term, the equation is eliminated. This tool can eliminate the overfitting polynomial solutions.

3.4 Multiple alternative equations

Consider the search for empirical equations. The space of algebraic equations can be built by repeated combination of three spaces determined by (1) the set of models that are directly fitted to data, (2) the set of transformations that generate new terms, and (3) the choice of tuples of terms to be used in the equations. BACON-1 uses (1) constant or linear fit, (2) transformations * and /, (3) pairs of terms, at least one of them derived from the dependent variable. EF uses (1) polynomial equations up to a degree determined by data, (2) transformations *, /, exp. log, sqrt, (3) pairs of terms such that one is derived solely from x, while the other uses occurrences of y.

As a general principle, in automated discovery the search control should follow the simple-first strategy. If two solutions enjoy comparable support by data, the simpler of the two should be preferred. It has less parameters that require empirical interpretation. Simplicity is determined by factors such as the number of parameters in the equation and the number of term transformations used. When the simplicity criteria cannot be reduced to a single scale, solutions can be only arranged into partially ordered simplicity classes.

Since the order of search in each simplicity class is arbitrary, the search driven by simple-first control should stop after examining all solutions in the simplicity class C in which it found the first acceptable solution, or in the case of partially ordered simplicity classes should examine all classes not more complex than class C. EF conducts the search with a fixed acceptance threshold normally set at Q=0.001 and a fixed polynomial degree determined by data. To avoid explosive search, the depth in the number of transformations has been also limited (to one transformation for each of x and y).

3.5 Equations for time as a function of angle, at fixed mass

Following BACON-3 method, Equation Finder (EF) has been applied for each mass from mass-1 to mass-9, to data in the format (angle, time, error of time). Because of the small number of 5 datapoints and their monotonicity, EF limits the search to linear fit. Repeated application of EF led to nine sets of equations, one for each data set. The same five equations out of 46 considered by EF have been consistently the best in each data set. We shall call them $eq_1 - eq_5$:

eq_1: $y = (A + B\sqrt{x})/x$ $\quad$ eq_2 $\; y = \exp(A + B \log x)$ $\quad$ eq_3 $\; y = 1/(A + B\sqrt{x})$
eq_4 $\; y = \log(A + B/x)$ $\quad$ eq_5 $\; y = \sqrt{A + B/x}$

These five equations have been accepted at least in six data sets each, but unaccepted in some data sets. The best five equations generated by EF for mass-2 are reproduced below in a slightly edited lisp form:

```
Y = (/ 1 (+ A (* B (SQRT X))))      (A B) = (-0.083 0.370) +- (0.016 0.004)     Chi2 = 8.77   Q = 0.033
Y = (/ (+ A (* B (SQRT X))) X)      (A B) = (0.732 2.680) +- (0.104 0.038)      Chi2 = 9.19   Q = 0.027
Y = (EXP (+ A (* B (LOG X))))       (A B) = (1.176 -0.545) +- (0.015 0.006)     Chi2 = 11.46  Q = 0.010
Y = (SQRT (+ A (/ B X)))            (A B) = (-0.0710 9.403) +- (0.0118 0.153)   Chi2 = 21.41  Q = 8.63d-5
Y = (LOG (+ A (/ B X)))             (A B) = (1.216 13.260) +- (0.018 0.232)     Chi2 = 30.10  Q = 1.31d-6
```

EF returns error for the value of each parameter. For instance, in the last equation, the value 1.216 of A has error 0.018, while the value 13.260 of parameter B has error 0.232. The first three have been accepted, since their probability Q has been higher than the threshold value $Q_a = 0.001$.

3.6 Generalization to mass

Different equations have been the winners for different masses and none has been acceptable for all masses. This motivated us to expand FAHRENHEIT by a module which prioritizes the equations and redefines their acceptability.

Even if no single form of equation is acceptable for all datasets, generalization should be attempted. The situation is similar to the evaluation of a single equation. Even if an occasional data point is at a distance greater than error from the value predicted by equation, the probability of fit can be still high and the fit can be accepted.

FAHRENHEIT has been expanded to consider each form of equation accepted in at least two thirds of datasets by EF, generalizes all such equations and computes the probability measure Q in the same way as for 2-D equations. For comparison, we also compute several other evaluation metrics. Consider the equations for mass 1–9, summarized below by three measures:

1. The total of ranks of a given equation type in nine sets of equations. For instance, equation-1 was the first twice (2pts), second three times (6pts), third twice (6pts), and fourth twice (8pts), for the total of 22pts.

2. Total χ^2 for all nine equations. The best measure of fit.

3. The ratio of datasets in which the equation was accepted to all datasets.

MEASURE:	1. RANK	2. CHI-2 TOTAL	3. ACCEPTANCE RATIO
eq_1	22	68.8	8/9
eq_2	24	65.3	8/9
eq_3	26	71.0	8/9
eq_4	31	104.4	6/9
eq_5	32	101.4	6/9

The results show that three equation types are distinctly better. Each of the three equations has been accepted at the probability threshold 0.001 for eight out of nine masses.

The following postulates follow from our analysis: (1) a few unacceptable fits, unless very bad (big value of χ^2) do not disqualify a given type of equation; (2) not accepted equations must be remembered at the first level, since some of them will be used at the next level if accepted in the majority of datasets.

The first three equations have been admitted to the next level of search, but for comparison we also used eq_4. Each equation has two parameters, A and B.

For each equation type, FAHRENHEIT creates two datasets, one for parameter A and one for B: $(m_i, A_i, \varepsilon_{A_i})$ and $(m_i, B_i, \varepsilon_{B_i})$. The value of error for each parameter value has been generated by EF at the first level. Now EF is applied to both datasets. The best fits are shown below for the best two equations. Notice, that the majority of searches have been successful without any term transformations.

```
eq_1:   time = (A + B * SQRT(angle)) / angle
For A:  A = (+ Aa (* Ab mass)) ,           (Aa Ab) = (0.248 0.0018) +- (0.1020 0.0005)    Chi2 = 15.0   Q = 0.036
For B:  B = (+ Ba (* Bb mass)) ,           (Ba Bb) = (3.016 -0.0026) +- (0.035 0.0002)    Chi2 = 7.7    Q = 0.361

eq_2 :  time = EXP(A + B * LOG(angle))
For A:  A = (LOG (+ Aa (* Ab mass mass))) , (Aa Ab) = (3.09  -2.9e-6) +- (0.03 6.4e-7)     Chi2 = 23.9   Q = 0.001
For B:  B = (+ Ba (* Bb mass)) ,           (Ba Bb) = (-0.509 -1.65e-4) +-(0.006 3.24e-5) Chi2 = 16.5   Q = 0.021
```

To produce equations of the form $t = g(angle, mass)$, the equations for A and B can be substituted into the original equation $t = f(angle)$.

4 Evaluation of the results

We shall now investigate the quality of the equations proposed by FAHRENHEIT.

4.1 Empirical equations vs. data

Total χ^2 $(\sum_{i=1}^{N}[(y_i - f(x_i))/\sigma_i]^2)$ and the corresponding probability Q summarize the fit of equation $y = f(x)$ to data (x_i, y_i, σ_i). Residua $f(x_i) - y_i$ provide detailed information about local areas of fit and misfit. The analysis of residua demonstrates no substantial areas of misfit between data and predicted values, leading to the conclusion that empirical equations provide a joint description of all phenomena that contributed to the data.

The total χ^2 for each type of equation is shown below. We computed χ^2 for each of the four final equations, and we also show the number obtained by totaling χ^2 for each of the nine equations at the first level (for mass). For each equation, both numbers offer answers to an important question: how much fit is lost in generalization from bi-variate equations derived on the first level to the equation derived at the second level. The χ^2 for the final equation must be bigger that the total of χ^2's for the first level equations because the values of A and B derived from equations at the second level cannot be as good as the best one, while the best fit has been assured for each equation on the first level.

Equation:	eq1	eq2	eq3	eq4	mechanics
chi-2 level1:	69	65	71	104	
chi-2 level2:	109	113	115	154	711

The total χ^2 values of the best equations show that the global fit of the theoretical equation (5) (see last column at level2, above) was far worse than that of the best empirical equations.

4.2 Galileo's theory vs. data

After the cylinder is released by the gripper and starts rolling down, the top touch sensor changes its state from engaged to released and the timer is turned on. When that happens, however, the cylinder has already rolled a short distance of about 0.7cm, and its initial velocity $V1$ is greater than zero. We haven't reduced this distance to zero as we wanted to keep a safety margin, so that the top sensor is always successfully engaged. Galileo's theory from section 2.1, applied to our setup, leads to the equation:

$$t = \sqrt{\frac{2D}{gsin(\alpha)}} - \sqrt{\frac{2d}{gsin(\alpha)}} \tag{3}$$

We used this equation to predict time for each of the nine cylinders at each angle. Since Galileo's theory does not capture rotational energy and momentum of inertia, the predicted values offer a very poor fit to data. The residua are at the level of about 25 percent of the empirical values, compared to about 1% of the prediction error of our best empirical equations. The original equation of Galileo, however, could do better, since the constant was determined from his data, eliminating the systematic underfit.

4.3 Theoretical equation vs. data

The following theoretical equation on time t of descent of a hollow cylinder of mass m, derived from classical physics, combines loss of potential energy, gain in kinetic energy of translation, and gain in the kinetic energy of rotation:

$$t = \sqrt{\frac{D(4M-m)}{sin(\alpha)gM}} \tag{4}$$

where D is the distance rolled by the cylinder, M is mass of a solid cylinder of the same diameter as the given cylinder (mass m), and g is Earth acceleration. From equation 4 we derive the theoretical equation that takes into account distance d that the cylinder covered before the top sensor was released

$$time = \sqrt{\frac{D(4M-m)}{sin(\alpha)gM}} - \sqrt{\frac{d(4M-m)}{sin(\alpha)gM}} \tag{5}$$

$D = 50.2cm$, $d = 0.7cm$. We used $g = 980.03cm/s^2$ received from physicists in our building.

The analysis of residua for the theoretical equation (5) suggests that additional factors, not accounted for in the theoretical model, systematically alter results in two areas. For the smallest angle the time predicted from (5) is systematically shorter than the time measured, while for larger angles, the time predicted from (5) is systematically longer. Cylinder sliding is a likely factor at higher angles. Energy dissipation by static friction is likely at the lowest angle.

5 Conclusions

Robotic systems that interact with the real world, gathering empirical data and developing theories, make an attractive long-term goal for machine discovery. Automation of discovery meets a demanding test when applied on robots.

Our reconstruction of Galileo's experiment shows that a simple discovery system linked to a simple robot arm can generate empirical equations that fit the data better than equations derived from a well-established theory. The influence of processes that are hard to represent in a theoretical model can be captured by empirical equations.

It may be futile to expect a robot discoverer to deliver a distinctly superior single solution, typically sought in machine discovery. For automated systems, single solutions are a desired but unlikely borderline between no solutions and many solutions. Automated discoverer may require plenty of further knowledge to gain the perspective necessary to make choices between competing equations and to move from empirical equations to basic theories.

References

Dzeroski, S. & Todorovski, L. 1993. Discovering Dynamics, *Proc. of 10th International Conference on Machine Learning*, 97-103.

Kulkarni, D., & Simon, H.A. 1987. The Process of Scientific Discovery: The Strategy of Experimentation, *Cognitive Science, 12*, 139-175.

Langley, P.W., Simon, H.A., Bradshaw, G., & Zytkow J.M. 1987. *Scientific Discovery; Computational Explorations of the Creative Processes.* Boston, MA: MIT Press.

Moulet, M. 1992. A symbolic algorithm for computing coefficients' accuracy in regression, in: Sleeman D. & Edwards P. eds. *Proc. 9th Int. Conference on Machine Learning.*

Nordhausen, B., & Langley, P. 1990. An Integrated Approach to Empirical Discovery. in: J.Shrager & P. Langley eds. *Computational Models of Scientific Discovery and Theory Formation*, Morgan Kaufmann Publishers, San Mateo, CA, 97-128.

Zembowicz, R. & Żytkow, J.M. 1991. Automated Discovery of Empirical Equations from Data. In Ras. Z. & Zemankova M. eds. *Methodologies for Intelligent Systems*, Springer-Verlag, 1991, 429-440.

Żytkow, J.M., Zhu, J. & Hussam, A. 1990. Automated Discovery in a Chemistry Laboratory, *Proceedings of the AAAI-90*, AAAI Press, 889-894.

Żytkow, J.M. 1996. Automated Discovery of Empirical Laws, *Fundamenta Informaticae, 27*, p.299-318.

...