

Dehydration of *t*-Butyl Alcohol Using Sulfonic Ion Exchange Resins

Frank J. Vilimek

Submitted: March 13, 2023

Abstract

The dehydration reaction of *t*-butyl alcohol to form isobutylene and water was studied to discern the effects of water on catalyst poisoning, and thus the effects on reaction rate involving sulfonic ion exchange resins. It was found that at high concentrations of *t*-butyl alcohol, around $10M$, the reaction proceeded quickly and produced high levels of water, which later poisoned the catalyst and decreased the reaction rate. For mixtures of *t*-butyl alcohol and water up to a water concentration of $5.55M$, the reaction rate experienced a sharp decrease as water began competing for catalyst sites; above a water concentration of $5.55M$, enough water had adsorbed to the catalyst that it acted as a catalyst itself, however weaker than the sulfonic sites. For mixtures of *t*-butyl alcohol and methylcyclohexane, the reaction proceeded relatively quickly, as methylcyclohexane did not poison the catalyst nearly as readily as water, allowing the dehydration reaction to proceed. Performing nonlinear curve fitting showed that water was 9.77 times more likely, on average, to adsorb to the catalyst than *t*-butyl alcohol, which is detrimental to the overall production of isobutylene, yet can be used in large-scale processes to operate in a less unstable regime where small deviations in water concentration can drastically change output levels.

Introduction

Isobutylene is a hydrocarbon monomer that is used as a precursor in the production of butyl rubber, a material often used in heat/weather insulation, rubber football bladders, and some food products like chewing gum [3]. When reacted with methanol, it forms methyl *t*-butyl ether (MTBE), a fuel additive that raises the octane number for smoother engine operation [2]. Having high industrial value, isobutylene must be produced using efficient methods with high yields.

Isobutylene can be produced using several reactions, such as the high-temperature back-cracking of MTBE to separate methanol and isobutylene [2], however this process also produces a mixture of C_4 molecules. These C_4 molecules, some of which include butane and butadiene [6], have similar boiling points to isobutylene, making separating isobutylene difficult without a distillation column with a large number of stages. To avoid purchasing and installing a large and expensive distillation column, isobutylene is instead widely-produced by the catalytic dehydration of *t*-butyl alcohol (TBA), which requires a stream of TBA and ion exchange resins [4]. This process is very efficient as the reaction only produces isobutylene and water, and does not require extensive purification steps after the reaction has taken place.

The dehydration of TBA using ion exchange resin was studied by Bruce Gates and William Rodriguez in their 1973 paper *General and Specific Acid Catalysis in Sulfonic Acid Resin* [1], wherein the reaction kinetics were analyzed in terms of the Langmuir-Hinshelwood mechanism. This study, which was the inspiration for our study, determined that the resin became poisoned by water produced in the dehydration reaction; the initial reaction rate experienced a significant drop once water reached a concentration greater than 5.6M (9M of TBA). They found that water adsorbed to the catalyst more readily than TBA.

Using a similar experimental setup and data acquisition techniques, our study aimed to recreate Gates' and Rodriguez's results using a different catalyst, labeled *Resin D*. We found the same poisoning effect of water on the catalyst, as well as the relative adsorption of water to TBA.

Experimental Methods

The lab was split up into two main portions: the first portion was dedicated to finding the number of catalyst sites per gram on our specific catalyst, and the second portion was dedicated to carrying out several reactions using solutions of varying concentration of TBA and water, or TBA and methylcyclohexane (MCH).

To find the number of catalyst sites per gram of catalyst, 50 mL of 0.1N NaOH solution was prepared with 1 gram of catalyst (*Resin D*) and 10 grams of NaCl. The solution was allowed to sit overnight and was later titrated using 0.1N HCl, which was used to calculate catalyst sites per 1 gram of catalyst.

After calculating the number of sites, the same type of catalyst was used in several TBA dehydration reactions in order to obtain initial reaction rate data. The day prior to each lab session, catalyst samples weighing between 0.2 and 3 grams were placed in a vacuum evaporator held at 393K to remove any water that may be adsorbed to the catalyst.

For each trial, a round-bottom flask was rinsed with a small amount of TBA before being filled with a solution of either TBA and water or TBA and MCH. A magnetic stir bar was set to 500 rpm and the flask was heated to 80°C. A sample of catalyst was added to the flask and a timer was started. The entrance to the flask was sealed and the volumetric flow rate was recorded using either the digital data acquisition software or timing gas evolution through bubble formation. After roughly 12 minutes the reaction was stopped, and the contents of the flask were dumped and the flask was rinsed using DI water and acetone. The flask setup and volumetric flow rate measurements were repeated for the remainder of the catalyst samples.

Theory and Analysis

Catalytic Dehydration of TBA

The catalytic dehydration of TBA ($((\text{CH}_3)_3\text{COH})$) involves the removal of the OH^- ion and one H^+ ion, forming one molecule of isobutylene ($((\text{CH}_3)_2\text{C}=\text{CH}_2)$) and one molecule of water. The ions are removed after TBA has adsorbed onto the surface of the catalyst, which is composed of a matrix of sulfonated poly(styrene-divinylbenzene), whose structure is seen in fig. (1). The resin is produced by crosslinking chains of styrene and divinylbenzene to form a porous polymer matrix, then reacting the polymer with concentrated sulfuric acid to attach $-\text{SO}_3\text{H}$ groups [8], which act as catalytic sites.

Once TBA adsorbs to the resin (catalyst), the OH^- and H^+ ions become strongly attached to the $-\text{SO}_3\text{H}$ group, allowing isobutylene to detach and enter the vapor phase. The water molecule

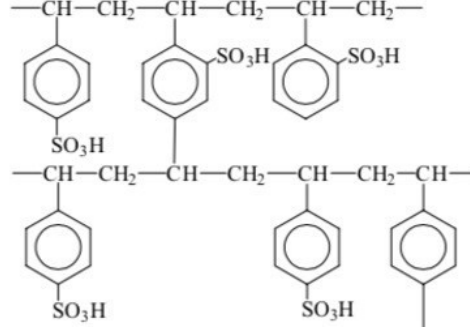


Figure 1: Sulfonated poly(styrene-divinylbenzene) structure, which forms a cationic resin [8].

stays adsorbed to the catalyst, hindering TBA from adsorbing to that site. Eventually, enough water adsorbed to the catalyst will hinder TBA adsorption and the production of isobutylene will be greatly decreased. Once enough water is present on the catalyst, catalysis will shift from the sulfonated sites to the hydrated protons of water, allowing the dehydration reaction to continue, yet at a slower rate.

Langmuir-Hinshelwood Mechanism

The adsorption of TBA and water can be described using the Langmuir-Hinshelwood mechanism, which models molecules adsorbing to a surface, reacting, and then desorbing. A representation of this is shown in fig. (2).

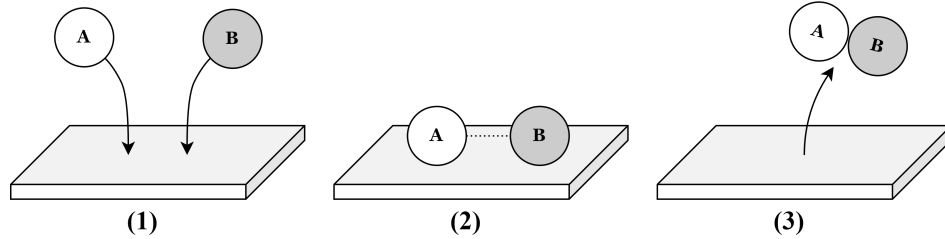


Figure 2: The Langmuir-Hinshelwood mechanism describes the adsorption (1), reaction (2), and desorption (3) of molecules, and is often used to model catalytic reactions.

Using this mechanism, the particular equation used to describe the production of isobutylene through the dehydration of TBA is given by

$$r_A = \frac{\kappa K_A C_A}{1 + K_A C_A + (K_W C_W)^\gamma} \quad (1)$$

where κ is a lumped reaction rate constant; K_A is the TBA adsorption parameter; C_A is the concentration of TBA; K_W is the water adsorption parameter; C_W is the concentration of water; and γ is a rate-order describing the adsorption of water.

There is also an observed pseudo 1st-order reaction instead of the Langmuir-Hinshelwood mechanism for high concentrations of water, given by the rate equation

$$r' = \kappa' C_A \quad (2)$$

where κ' is the pseudo 1st-order reaction rate constant. Combining eq. (1) and eq. (2) yields the entire rate equation that we would expect the reaction to follow

$$r = r_A + r' = \frac{\kappa K_A C_A}{1 + K_A C_A + (K_W C_W)^\gamma} + \kappa' C_A \quad (3)$$

In order to find these reaction rate parameters, estimates can first be found by using limiting cases: for high concentrations of water, we are left with eq. (2), allowing us to fit a line to find κ' through the slope; for very high concentrations of TBA, the reaction rate approaches the value of κ ; for very low concentrations of water, the rate equation becomes

$$r = \frac{\kappa K_A C_A}{1 + K_A C_A} \quad (4)$$

allowing us to fit a curve to find K_A ; finally, with the rest of the parameters calculated, K_W can be found through curve-fitting. These estimates will serve as initial guesses for more extensive nonlinear curve-fitting, which will also be used to determine the optimal value for γ .

Results and Discussion

Initial Rates from Volumetric Flows

The volumetric flow rate of isobutylene out of the reactor was measured using both the digital and bubble flow meters. The linear portion of their flow rates was used to extrapolate backwards in time to estimate the initial reaction rate. Figure (3) shows two examples of this extrapolation – the intercept of the linear regression is the initial reaction rate, which was normalized by the concentration of catalyst sites per gram of catalyst. The concentration of catalyst sites was found to be $C_t = 0.0056 \pm 0.0017$ sites/gram through titration.

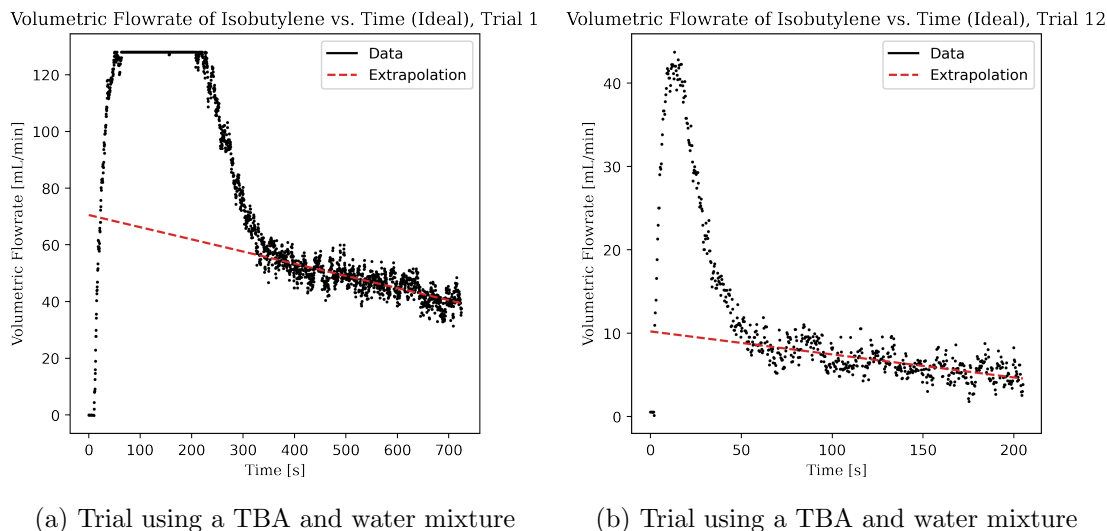


Figure 3: Volumetric flow rates were used to extrapolate backwards in time to estimate the initial reaction rate.

The initial reaction rates were estimated assuming that isobutylene behaved as an ideal gas and that isobutylene was the only species exiting through the flow meter. The initial reaction rates for

each trial were plotted against their TBA concentration, and this data was used to fit eq. (3) to find the parameters κ , κ' , K_A , K_W , and γ .

Fitting Langmuir-Hinshelwood to Data

The rate-versus-concentration data was fitted using `scipy.optimize.curve_fit` in Python. The parameters were found using limiting cases and are listed in table (1).

Table 1: Reaction rate parameters fitted to different γ -values. Errors are given as 1σ values, obtained by propagating equipment error through calculations.

γ	κ	K_A	K_W	κ'	SSE
	[mol/equiv. acid grps sec]	[Liter/mol]	[Liter/mol]	[L/equiv. acid grps sec]	[mol ² /equiv. acid grps ² sec ²]
1	0.04065 \pm 0.00418	0.3975 \pm 0.00217	5.0 \pm 0.871	0.000429 \pm 3.85e-5	0.000109
2	0.04065 \pm 0.00418	0.3975 \pm 0.00217	3.883 \pm 2.102	0.000429 \pm 3.85e-5	0.000105
2.93	0.04065 \pm 0.00418	0.3975 \pm 0.00217	1.836 \pm 0.785	0.000429 \pm 3.85e-5	0.000104
3	0.04065 \pm 0.00418	0.3975 \pm 0.00217	1.836 \pm 0.785	0.000429 \pm 3.85e-5	0.000107

These parameters were used to plot eq. (3) to the data; the integer value for γ that gives the smallest sum-of-squares error (SSE) is $\gamma = 2$, and the line fit is shown in fig. (4). The curve fit more clearly shows that the initial reaction rate experiences a sharp decrease for even low water concentrations, which tapers off after water reaches a concentration of about 5.55M. This sharp decrease in the reaction rate for low water concentrations indicates the region where water competes with TBA for catalyst sites; since water is a very concentrated liquid, even small amounts of it can compete with high concentrations of TBA. When the concentration of water exceeds 5.55M, the reaction rate tapers off, indicating the region where enough water has adsorbed to the catalyst and are now acting as the catalyst sites themselves.

The TBA and MCH curve represents the trend of decreasing TBA concentration in the absence of another species that competes for catalyst sites. Even for relatively low concentrations of TBA in MCH, the reaction rate is close to an order of magnitude larger than those with low concentrations of TBA in water. This suggests that a decrease in reaction rate is not only due to decreasing TBA concentration, but also the choice of liquid mixed with TBA.

The non-integer value for γ that gives the lowest SSE was $\gamma = 2.93$; the figure is located in Appendix B. Non-integer values for γ could indicate that there is a more complex phenomenon occurring that our Langmuir-Hinshelwood expression could not capture. These γ -values are related to the number of water molecules adsorbing to the catalyst relative to the number of TBA molecules adsorbing to the catalyst; the Langmuir-Hinshelwood mechanism assumes that adsorption occurs on Langmuir adsorption isotherms, which in turn assumes that the catalyst has an even distribution of catalyst sites and they are all equally accessible by all adsorbates [7]. This is an idealized scenario,

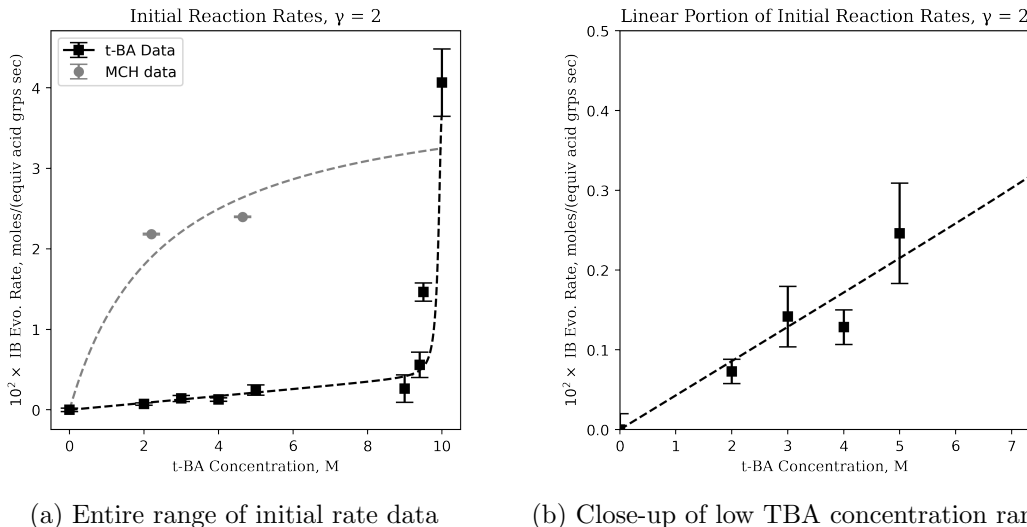


Figure 4: Initial reaction rate data fitted using $\gamma = 2$. The TBA-water curve was fitted using eq. (3), and the TBA-MCH curve was fitted using eq. (4).

and leads to integer values for γ for ideal reactions – since our catalyst and reaction are not ideal, the catalyst may have regions of varying concentrations of sites, making the ratio of water-to-TBA adsorption deviate from integer values.

The ratio of K_W to K_A represents how readily water adsorbs to the surface relative to TBA: for $\gamma = 2$, this ratio is 9.77 ± 5.29 . An interpretation of this is that water is 9.77 times more likely on average to adsorb to a catalyst site than TBA, which could explain the sudden drop in the initial reaction rate once water reaches some “critical” concentration. This is not without a significant amount of error however, and this may be due to attempting to fit a complex equation to 9 data points. It may also indicate that different samples of catalyst varied in site concentration or surface geometry, making it difficult for molecules to adsorb to.

Comparison to Literature Values

Gates’ and Rodriguez’s paper is the closest study to ours, and therefore we looked to it to compare reaction rate parameters. One parameter that should be similar across both studies is κ' , since for such high concentrations of water the choice of catalyst should have a negligible effect. For $\gamma = 2$, the value for κ' in their paper was $0.000184 \text{ L}/(\text{equiv. acid grps sec})$, which is less than half of our value of $0.000429 \text{ L}/(\text{equiv. acid grps sec})$. This discrepancy may be due to fitting the Langmuir-Hinshelwood equation to a small sample of points, which is likely not representative of the actual mechanism. It may also be due to systematic error, as the low TBA concentration range in fig. (4b) seems to follow the curve fit to a fairly close approximation.

Comparison with Gates’ and Rodriguez’s catalyst can also be done under the scenario of a large-scale isobutylene production process. The efficacy of a catalyst in this scenario is heavily dependent on how quickly the catalyst will inevitably get poisoned by water, and can be quantified by the ratio of K_W to K_A mentioned previously. The $K_W:K_A$ ratio in their paper was 10.05, however the error in these values was not listed. Their ratio indicates that water is 10.05 times more likely on average to adsorb to a catalyst site over TBA, and is larger than our ratio. In this case, our catalyst would allow less water to adsorb relative to TBA than Gate’s and Rodriguez’s catalyst, meaning that more isobutylene and water can be produced before poisoning the catalyst

and reaching the “critical” concentration. For a large-scale process, our catalyst would be favored for its longer effective lifetime.

Validity of Assumptions

Several assumptions were made in analyzing our data: (1) the reactor was perfectly mixed; (2) any TBA produced will go immediately into the vapor phase; (3) no other species entered the vapor phase; (4) isobutylene acts as an ideal gas; and (5) the reactor operated isothermally.

The perfect-mixing assumption is valid given the set stir rate: for 50 mL of solution, a ~ 1 -in stir bar was set to 500 rpm, which ensured that there were no regions of stagnant or poorly-mixed fluid, especially when considering the relative size of the stir bar to the volume of fluid. This allows us to measure the reaction rate without accounting for mass transfer limitations.

The immediate vapor-phase assumption and the assumption that no other species went into the vapor phase are interrelated, as the boiling point of isobutylene is much lower than that of TBA and water (-6.9°C compared to 83°C and 100°C , respectively [9, 10, 11]). When the reactor is heated to 80°C , any isobutylene that is produced almost immediately goes into the vapor phase and enter the condenser. Any TBA or water that evaporated will come into contact with the condenser (held at 5°C) and condense, refluxing back into the reactor. If isobutylene came into contact with the condenser, the temperature of the condenser would still be too high to fully condense it. This allows us to approximate the volumetric flow rate through the flow meter as pure isobutylene.

To verify the ideal gas assumption, the compressibility factor for isobutylene was calculated for each trial. Using the critical temperature and critical pressure for isobutylene [9], the compressibility factor was never found to be less than 0.98 – this implies that assuming ideal gas behavior accounted for less than 2% error in the volumetric flow rate measurements, and thus less than 2% error in the initial reaction rate calculation. This 2% error is much less than the error due to equipment noise for example, allowing us to verify the validity of this assumption.

The isothermal assumption was found to be invalid. Throughout several trials, the hot plate failed to maintain the 80°C setpoint and dropped to temperatures as low as 73°C . It is presumed that the PID controller in the hotplate may have malfunctioned, or the metal heat sink surrounding the hot plate was not able to transfer heat at the proper rate. Failure to keep isothermal conditions may invalidate trial data in some cases, however a deeper analysis may be required to verify the extent at which this occurs.

Scale-Up Process

The scale-up for this reaction is well-documented, as it is currently carried out across the industry. One design comes from the U.S. Patent from 1983 [12], and the process is much the same, however some separation steps are needed for this to be a continuous process: TBA solution enters a packed-bed reactor (PBR) and is dehydrated to form isobutylene and water. The isobutylene, water, and any unreacted TBA are sent to a distillation column, where it is expected that close to 99% of the isobutylene will evaporate off to be collected downstream. The leftover water and TBA can either be directly recycled back into the reactor or sent to another distillation column to separate the TBA, which gets recycled back into the reactor, and water, which gets sent out a purge stream. Figure (5) shows a general process flow diagram for this setup, however slightly modified from the patented version.

The reactor is operated at 120°C and 20 bar, ensuring that the reaction rate is sped up without evaporating the TBA. Roughly 40 liters of catalyst are packed into the reactor, and 4,000 kg/hr of TBA and 1,000 kg/hr of water are pumped into the reactor – this operates at a relatively high

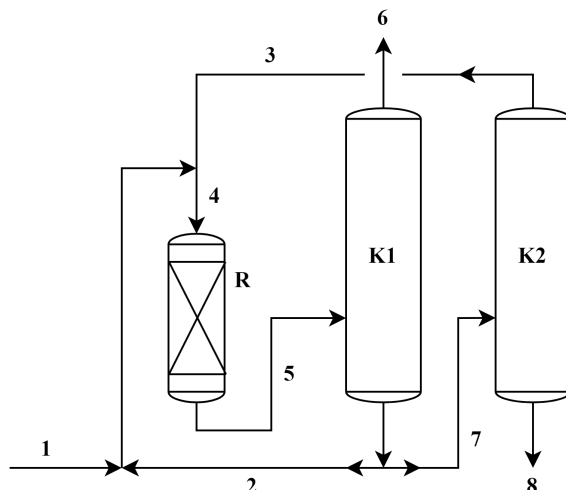


Figure 5: Example PFD for a large-scale TBA dehydration reaction, modified from a U.S. patented design. TBA and water enter the reactor R via stream 1, water exits out of stream 8, and TBA gets recycled back into the separator via stream 3. Distillation column K1 separates isobutylene from TBA and water, and distillation column K2 separates TBA from water.

concentration of water, however this leads to a more consistent product stream. For this particular setup, the isobutylene selectivity is expected to be $< 99.9\%$, and the isobutylene purity is also $< 99.9\%$. Though this method does require the use of distillation columns, the distillation of isobutylene and TBA from water is much easier than attempting to distill isobutylene from other C_4 hydrocarbons.

Conclusions and Recommendations

The dehydration of TBA using sulfonated ion exchange resins was studied by carrying out the reaction with varying weights of resin (catalyst) and TBA concentration in either water or MCH. It was found that TBA is dehydrated to form isobutylene at relatively quick reaction rates for high concentrations of TBA, however as the reaction progresses the catalyst begins getting poisoned by water. After a certain critical concentration of water is reached, the reaction proceeds at a fraction that of high TBA concentration reactions. The kinetics behind this reaction were described using the Langmuir-Hinshelwood mechanism, which accounts for the inhibitive effects of water adsorbing to the catalyst. Fitting the Langmuir-Hinshelwood expression to data using nonlinear curve fitting yielded adsorption parameters which indicate that water is 9.77 times more likely to adsorb to the catalyst than TBA, explaining the strong effect of the concentration of water in solution. When comparing our catalyst *Resin D* to the one used by Gates and Rodriguez, *Resin D* had the smaller $K_W:K_A$ ratio on average, meaning that it could operate for longer before water poisoning becomes an issue. The results of our experiment coincide with those found by Gates and Rodriguez, and the principles behind which can be used to design large-scale isobutylene production plants.

References

- [1] B.C. Gates, William Rodriguez, *General and specific acid catalysis in sulfonic acid resin*, Journal of Catalysis, Volume 31, Issue 1, 1973, Pages 27-31, ISSN 0021-9517, [https://doi.org/10.1016/0021-9517\(73\)90266-2](https://doi.org/10.1016/0021-9517(73)90266-2). (<https://www.sciencedirect.com/science/article/pii/0021951773902662>)
- [2] *High-Purity Isobutylene Production by MTBE Cracking*. S&P Global. (n.d.). Retrieved March 13, 2023, from <https://www.spglobal.com/commodityinsights/en/ci/products/chemical-technology-pep-reviews-high-purity-isobutylene-mtbe-cracking-2012.html>
- [3] *Isobutylene*. American Chemical Society. (n.d.). Retrieved March 4, 2023, from <https://www.acs.org/molecule-of-the-week/archive/i/isobutylene.html>
- [4] *Isobutylene*. (n.d.). Retrieved March 13, 2023, from <https://www.chemeurope.com/en/encyclopedia/Isobutylene>
- [5] Jain, A., Bansal, R., Kumar, A., & Singh, K. D. (2015). *A comparative study of visual and auditory reaction times on the basis of gender and physical activity levels of medical first year students*. International journal of applied & basic medical research. Retrieved March 5, 2023, from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4456887/>
- [6] Kent, J. A. &, Riegel, E. R. (2007). In *Kent and Riegel's Handbook of Industrial Chemistry and Biotechnology* (pp. 350–350). essay, Springer US.
- [7] *The Langmuir adsorption model*. Materials Characterization Lab. (2021, November 9). Retrieved March 13, 2023, from <https://mcl.mse.utah.edu/the-langmuir-adsorption-model/>
- [8] Meryt. “Ion Exchange Resins: Absolutely Fascinating.” MERYT, 1 Apr. 2021, <https://meryt-chemical.com/en/ion-exchange-resins-absolutely-fascinating/>.
- [9] National Institute of Standards and Technology. (n.d.). 1-propene, 2-methyl. 1-Propene, 2-methyl-. Retrieved March 13, 2023, from <https://webbook.nist.gov/cgi/cbook.cgi?ID=C115117&Mask=4>
- [10] National Institute of Standards and Technology. (n.d.). 2-propanol, 2-methyl. 2-Propanol, 2-methyl-. Retrieved March 13, 2023, from <https://webbook.nist.gov/cgi/cbook.cgi?ID=75-65-0>
- [11] National Institute of Standards and Technology. (n.d.). Water. Retrieved March 13, 2023, from <https://webbook.nist.gov/cgi/cbook.cgi?ID=C7732185>
- [12] “US4423271A - Process for Producing High Purity Isobutene by Dehydrating Tertiary Butanol.” Google Patents, Google, <https://patents.google.com/patent/US4423271A/en>.

A Appendix - Tables of Compressibility Factors and Errors

Table 2: Compressibility factors for each trial. The critical temperature used was $T_c = 417.9$ K, critical pressure was $P_c = 40 \times 10^5$ Pa [9]. Compressibility factors and error were not identical past the decimal points shown.

Trial #	Compressibility Factor z
1	0.982 ± 0.0016
3	0.983 ± 0.0017
5	0.983 ± 0.0017
8	0.983 ± 0.0017
10	0.983 ± 0.0017
11	0.983 ± 0.0017
12	0.984 ± 0.0017
13	0.984 ± 0.0017

Table 3: Error in initial reaction rate due to error using the digital flow meter and the bubble flow meter. Bubble flow meter error was quantified by using the average visual reaction time of ~ 0.25 seconds [5], which is the lower bound for error.

Trial #	Digital Error	Bubble Error
	[mol/equiv. acid grp sec]	[mol/equiv. acid grp sec]
1	0.0041	0.0107
3	0.0017	0.0004
5	0.0015	0.0010
8	0.0011	0.0144
10	0.0006	0.0090
11	0.0004	0.0046
12	0.0002	0.0069
13	0.0002	0.0025

B Appendix - Additional Figures

B.1 Volumetric Flow Rate Data

Volumetric Flowrate of Isobutylene vs. Time (Ideal), Trial 3

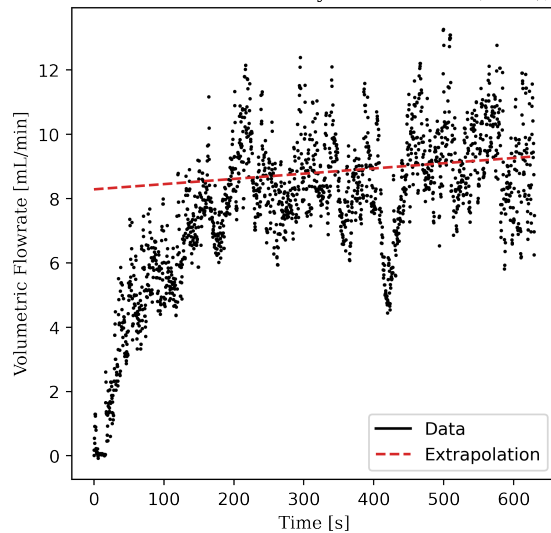


Figure 6: TBA-water reaction volumetric flow rate, using the digital flow meter.

Volumetric Flowrate of Isobutylene vs. Time (Ideal), Trial 5

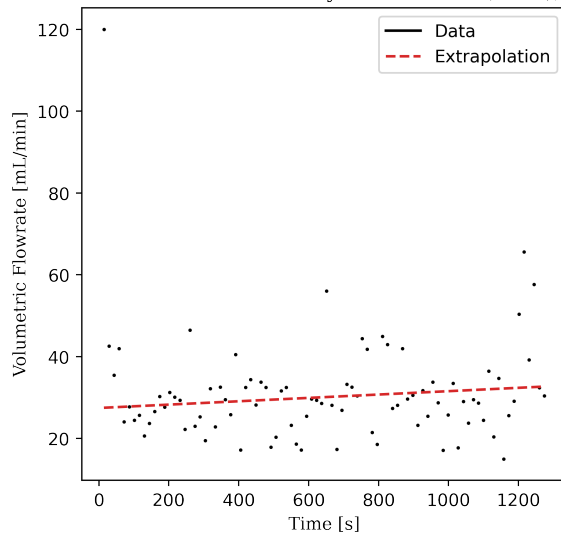


Figure 7: TBA-water reaction volumetric flow rate, using the bubble flow meter.

Volumetric Flowrate of Isobutylene vs. Time (Ideal), Trial 8

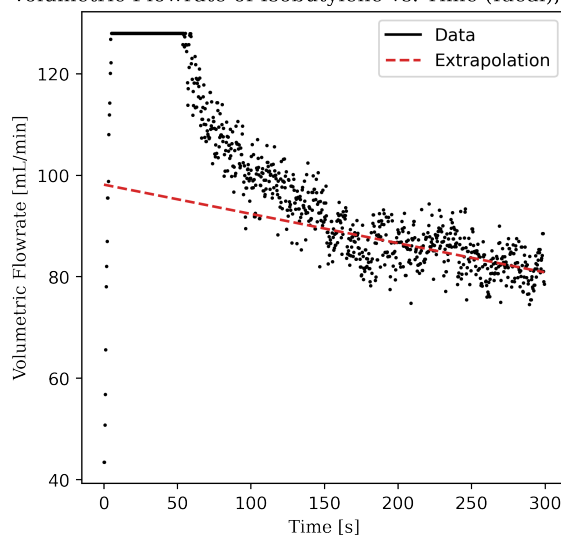


Figure 8: TBA-water reaction volumetric flow rate, using the digital flow meter.

Volumetric Flowrate of Isobutylene vs. Time (Ideal), Trial 10

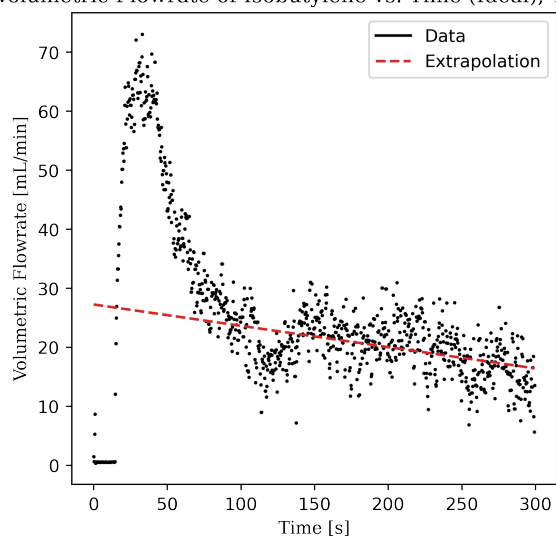


Figure 9: TBA-water reaction volumetric flow rate, using the digital flow meter.

Volumetric Flowrate of Isobutylene vs. Time (Ideal), Trial 11

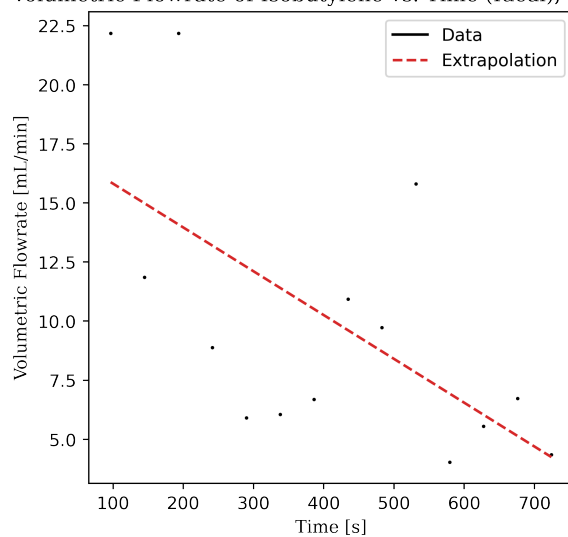


Figure 10: TBA-water reaction volumetric flow rate, using the bubble flow meter.

Volumetric Flowrate of Isobutylene vs. Time (Ideal), Trial 13

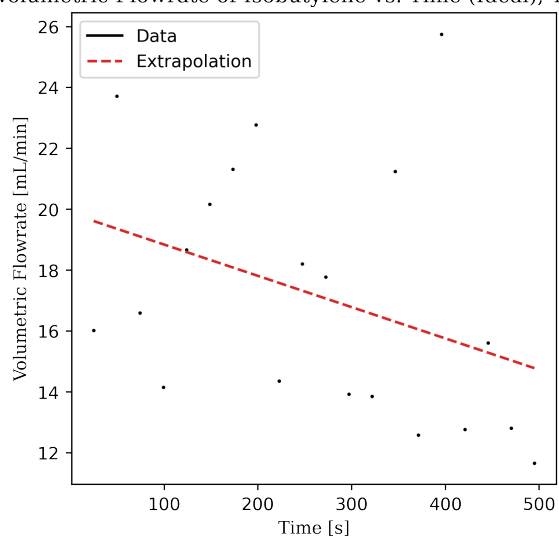


Figure 11: TBA-water reaction volumetric flow rate, using the bubble flow meter.

Volumetric Flowrate of Isobutylene vs. Time (Ideal), MCH 1

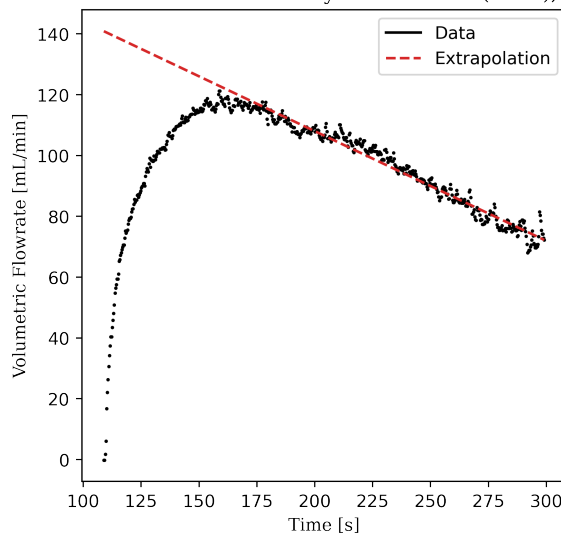


Figure 12: TBA-MCH reaction volumetric flow rate, using the digital flow meter.

Volumetric Flowrate of Isobutylene vs. Time (Ideal), MCH 2

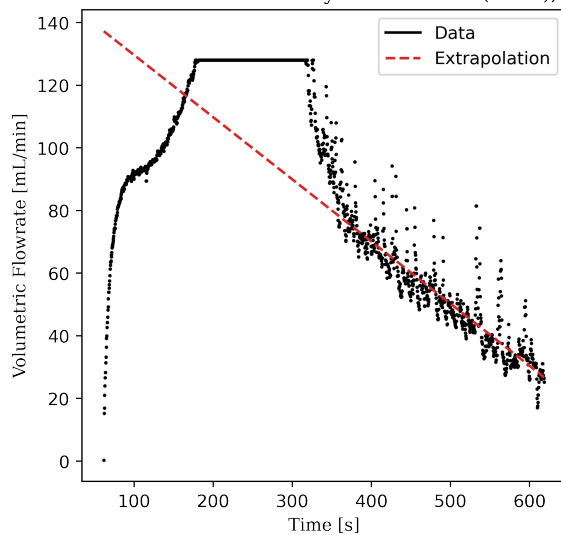


Figure 13: TBA-water reaction volumetric flow rate, using the digital flow meter.

B.2 Langmuir-Hinshelwood Equation Fitted with Varying γ

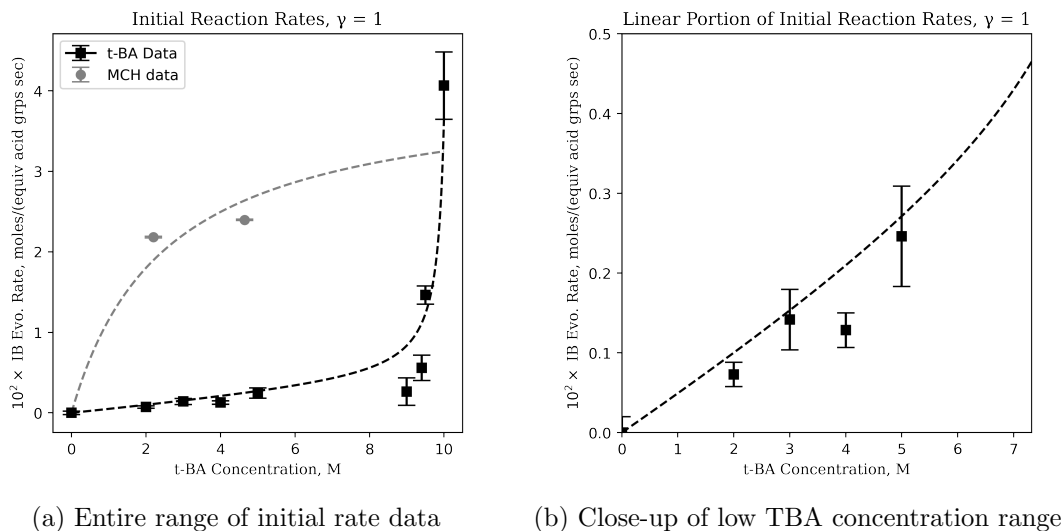


Figure 14: Initial reaction rate data fitted using $\gamma = 1$.

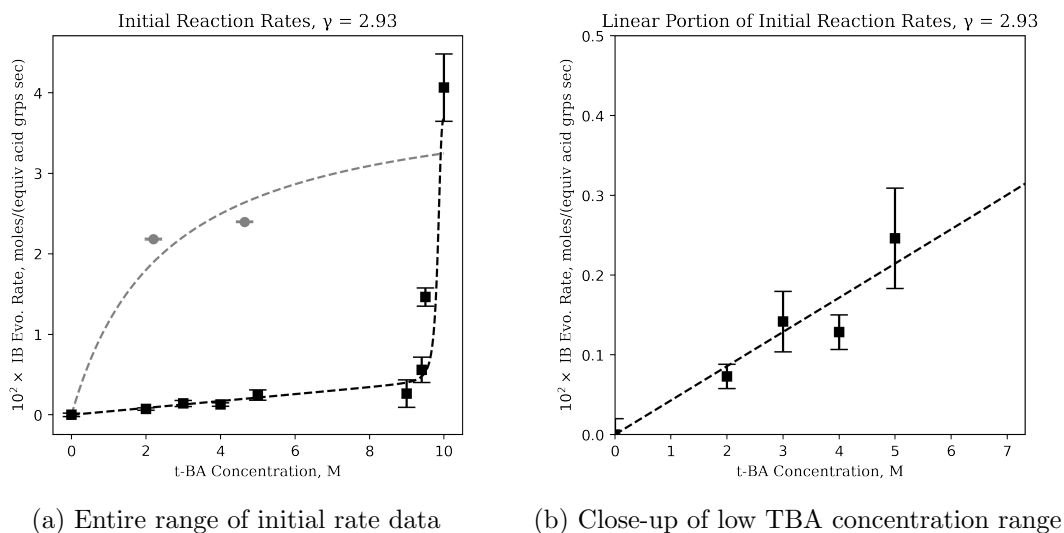
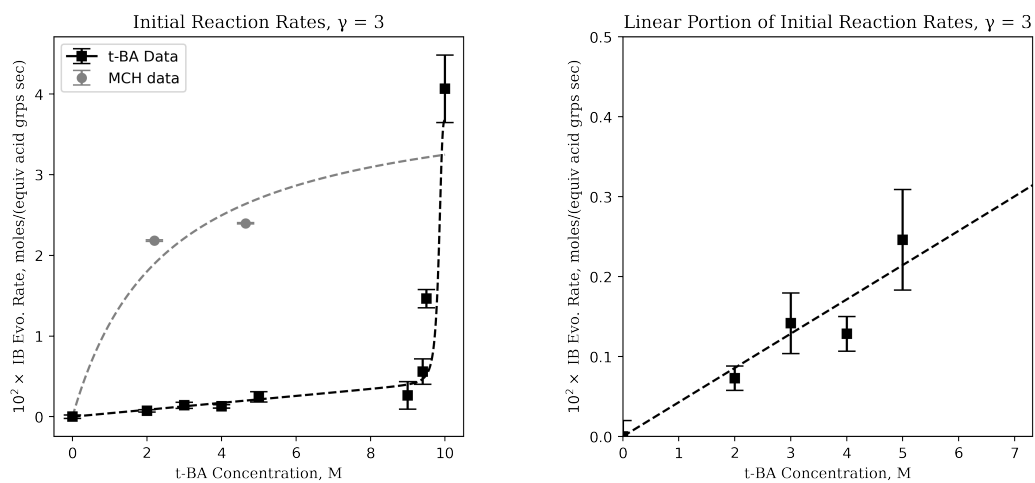


Figure 15: Initial reaction rate data fitted using $\gamma = 2.93$.



(a) Entire range of initial rate data

(b) Close-up of low TBA concentration range

Figure 16: Initial reaction rate data fitted using $\gamma = 3$.

Alternative Scale-Up Designs

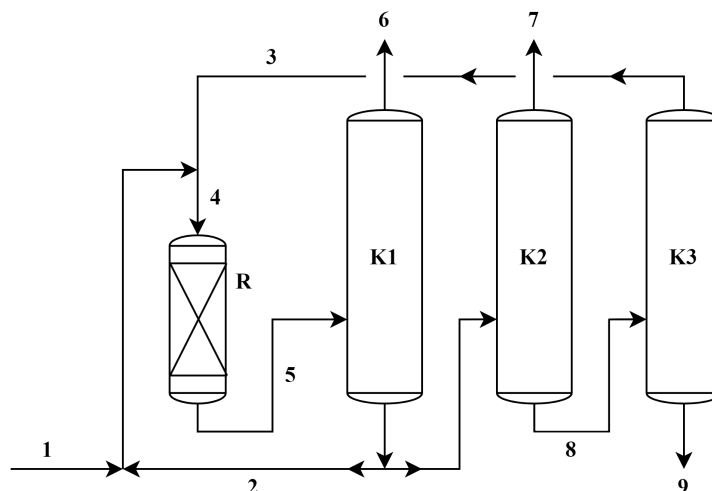


Figure 17: TBA dehydration scale-up design. This design is most similar to the patented design, utilizing a third distillation column for a more extensive purification of TBA before it gets recycled.

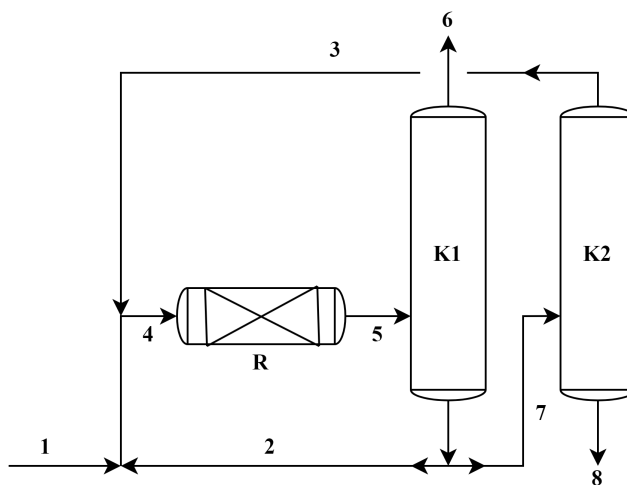


Figure 18: TBA dehydration scale-up design. This design utilizes a horizontal PBR instead of an upright PBR to allow isobutylene to flow more easily from one end to the other.

C Appendix - Error Analysis

C.1 Error in the number of moles (using Ideal Gas Law)

The error in the number of moles, assuming that the Ideal Gas Law is valid, can be found by first taking the derivative of the ideal gas equation, $n = PV/RT$, with respect to each of its variables

$$\frac{\partial n}{\partial P} = \frac{V}{RT} \quad \frac{\partial n}{\partial V} = \frac{P}{RT} \quad \frac{\partial n}{\partial T} = -\frac{PV}{RT^2} \quad (5)$$

We can plug these expressions into the master error propagation equation to yield

$$\sigma_n = \sqrt{\left(\frac{V}{RT}\right)^2 (\delta_P)^2 + \left(\frac{P}{RT}\right)^2 (\delta_V)^2 + \left(-\frac{PV}{RT^2}\right)^2 (\delta_T)^2} \quad (6)$$

C.2 Error in the van der Waals E.O.S. parameters

The van der Waals equation of state is composed of two parameters a and b that depend on the critical temperature T_c and the critical pressure P_c . These can be used to find the compressibility factor later on. The error in a and b is found by propagating error in the values for T_c and P_c , which were found on the NIST website. The derivatives of a and b with respect to T_c and P_c must first be found

$$a = \frac{27R^2T_c^2}{64P_c} \Rightarrow \frac{\partial a}{\partial T_c} = \frac{27R^2T_c}{32P_c} \quad \text{and} \quad \frac{\partial a}{\partial P_c} = \frac{-27R^2T_c^2}{64P_c^2} \quad (7)$$

$$b = \frac{RT_c}{8P_c} \Rightarrow \frac{\partial b}{\partial T_c} = \frac{R}{8P_c} \quad \text{and} \quad \frac{\partial b}{\partial P_c} = \frac{-RT_c}{8P_c^2} \quad (8)$$

The error in a is given by

$$\sigma_a = \sqrt{\left(\frac{27R^2T_c}{32P_c}\right)^2 (\delta_{T_c})^2 + \left(\frac{-27R^2T_c^2}{64P_c^2}\right)^2 (\delta_{P_c})^2} \quad (9)$$

and the error in b is given by

$$\sigma_b = \sqrt{\left(\frac{R}{8P_c}\right)^2 (\delta_{T_c})^2 + \left(\frac{-RT_c}{8P_c^2}\right)^2 (\delta_{P_c})^2} \quad (10)$$

C.3 Error in the compressibility factor

The compressibility factor z is found using a modified version of the van der Waals equation of state

$$z = \frac{V_m}{V_m - b} - \frac{a}{RTV_m} \quad (11)$$

where V_m is the molar volume of a species. Taking the derivative of z with respect to V_m , a , b , and T yields

$$\frac{\partial z}{\partial V_m} = -\frac{b}{(V_m - b)^2} + \frac{a}{RTV_m^2}; \quad \frac{\partial z}{\partial a} = -\frac{1}{RTV_m}; \quad \frac{\partial z}{\partial b} = \frac{V_m}{(V_m - b)^2}; \quad \frac{\partial z}{\partial T} = \frac{a}{RT^2V_m} \quad (12)$$

The error in the molar volume is given by

$$\delta_{V_m} = \sqrt{\left(\frac{1}{n}\right)^2 (\delta_V)^2 + \left(-\frac{V}{n^2}\right)^2 (\delta_n)^2} \quad (13)$$

The error in the compressibility factor is thus given by

$$\begin{aligned} \sigma_z = & \left[\left(-\frac{b}{(V_m - b)^2} + \frac{a}{RTV_m^2} \right)^2 (\delta_{V_m})^2 + \left(-\frac{1}{RTV_m} \right)^2 (\delta_a)^2 \right. \\ & \left. + \left(\frac{V_m}{(V_m - b)^2} \right)^2 (\delta_b)^2 + \left(\frac{a}{RT^2V_m} \right)^2 (\delta_T)^2 \right]^{\frac{1}{2}} \end{aligned} \quad (14)$$

C.4 Error in initial rate due to sensor noise

The error in the initial reaction rate due to sensor noise does not require propagating the error from several sources of error. Instead, it can be obtained by converting the standard error $\hat{\sigma}$ in its value (calculated with `scipy.optimize.curve_fit`) to a standard deviation σ using the formula

$$\sigma = \sqrt{n} \hat{\sigma} \quad (15)$$

where n is the total number of data points within a data set.

C.5 Error in initial rate due to human reaction time

The error in the initial reaction rate due to human reaction time can be calculated by first considering that the initial rate was found using the y -intercept of a line. We can thus find the error in the initial reaction rate by finding the error in the intercept of a line due to error in x

$$b = y - mx \quad \Rightarrow \quad \frac{\partial b}{\partial x} = -m \quad (16)$$

The error in the initial reaction rate is given by

$$\sigma_b = m \delta_x \quad (17)$$

D Appendix - Python Script for Calculations and Figures

```
#-----
#
# ECH 155 Lab 5 - TBA
#
# Group 2, Section A01: Arianna Corpuz, Kayla Naldo, Stephanie-Elliott Starks, Frank Vilimek
#
#-----

# Import libraries
import numpy          as np
import matplotlib.pyplot as plt
import pandas         as pd
import scipy.optimize as so
import scipy.stats    as ss
#import pint

font = {"fontname" : "Serif"} # Fancy font for plotting

# Accessing Files from Google Drive
from pydrive.auth      import GoogleAuth
from pydrive.drive     import GoogleDrive
from google.colab      import auth, files
from oauth2client.client import GoogleCredentials
!pip install sklearn # For R^2 calculation

auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)

# Pull CSV data from this link
link1 = ["https://drive.google.com/file/d/1LCGS2HGMDAbRGfa6IcvhJH1d-pyBzSA/view?usp=share_link",
         "https://drive.google.com/file/d/10wjK3sdggwYgemjM832cMic90qYyzrHS/view?usp=share_link",
         "https://drive.google.com/file/d/1TsFRMLx00u8DcgiFE3pvRSZKAuWL-qS2/view?usp=share_link",
         "https://drive.google.com/file/d/10edWWRqjD8oFozgMIMo1Pb7AVhV10hIx/view?usp=share_link",
         "https://drive.google.com/file/d/1p7ESxFpU6JKD7m1zF-2wtaAizFE5IFxm/view?usp=share_link",
         "https://drive.google.com/file/d/1r0_MVfyyta0cISBVGM0caCTODgJD1zXy/view?usp=share_link",
         "https://drive.google.com/file/d/1gZzXwffqfONOKOE0-NdupM87PpwAu_k0/view?usp=share_link",
         "https://drive.google.com/file/d/1rbAgNr0g8MMtFYoeww6pi0foHGBJFhNz/view?usp=share_link",
         "https://drive.google.com/file/d/1vRFWDaw0ogZTYwnIEkTOPMdLXMLV08qp/view?usp=share_link",
         "https://drive.google.com/file/d/1N3WZ5WAKDeMBfCTHULi3G_zEWj7U8uBM/view?usp=share_link",
         "https://drive.google.com/file/d/16TkGXmMzIFPsDdrTzYC242KVe093iTXy/view?usp=share_link",
         "https://drive.google.com/file/d/1KCIDvGotChbT8PUQQjNtjBn3lzEezUGn/view?usp=share_link",
         "https://drive.google.com/file/d/183ZraqNndhF59c3K96WxIrGfLjf8erBa/view?usp=share_link"]

link2 = ["https://drive.google.com/file/d/1_bwARA11PpCgBQsEZ4qBek6F1mviDDdz/view?usp=share_link",
         "https://drive.google.com/file/d/1KUxIE7_c9K7G8zi2WcQX5IP0JMqn-OyC/view?usp=share_link",
         "https://drive.google.com/file/d/1bIyVmdJdpHdR51sthNLtG_uuzcQ-bakK/view?usp=share_link"]

for i in range(len(link1)):
    id = link1[i].split("/")[-2]

    downloaded = drive.CreateFile({'id':id})
    downloaded.GetContentFile("Trial {}".format(i + 1))

for j in range(len(link2)):
    id = link2[j].split("/")[-2]

    downloaded = drive.CreateFile({'id':id})
    downloaded.GetContentFile("MCH {}".format(j + 1))

# TBA and water data
trial_dictionary = {"Trial 1" : "/content/Trial 1",
                   "Trial 3" : "/content/Trial 3",
```

```

        "Trial 5" : "/content/Trial 5",
        "Trial 8" : "/content/Trial 8",
        "Trial 10" : "/content/Trial 10",
        "Trial 11" : "/content/Trial 11",
        "Trial 12" : "/content/Trial 12",
        "Trial 13" : "/content/Trial 13"
    }

# Catalyst weights (grams)
cat_wt = {"Trial 1" : 0.215,
        "Trial 3" : 0.386,
        "Trial 5" : 0.597,
        "Trial 8" : 0.818,
        "Trial 10" : 1.340,
        "Trial 11" : 1.511,
        "Trial 12" : 1.689,
        "Trial 13" : 1.875
    }

# TBA concentrations (mol/L)
tba_conc = {"Trial 1" : 10.0,
        "Trial 3" : 9.0,
        "Trial 5" : 9.4,
        "Trial 8" : 9.5,
        "Trial 10" : 5.0,
        "Trial 11" : 3.0,
        "Trial 12" : 2.0,
        "Trial 13" : 4.0,
        "Zero" : 0.0
    }

# Water concentrations (mol/L)
h2o_conc = {"Trial 1" : 0.00,
        "Trial 3" : 5.55,
        "Trial 5" : 3.33,
        "Trial 8" : 2.775,
        "Trial 10" : 27.75,
        "Trial 11" : 38.85,
        "Trial 12" : 44.4,
        "Trial 13" : 33.3,
        "Zero" : 55.5
    }

# Data indices where data is approximately linear, found by inspection
linear_indices = {"Trial 1" : 1140,
        "Trial 3" : 1420,
        "Trial 5" : 1,
        "Trial 8" : 468,
        "Trial 10" : 392,
        "Trial 11" : 0,
        "Trial 12" : 258,
        "Trial 13" : 0
    }

# Moles of equivalent acid groups per gram catalyst
mol_site_per_g = 5.613E-3
error_sites = 1.665E-4

# MCH data
mch_data = {"MCH 1" : "/content/MCH 1",
        "MCH 2" : "/content/MCH 2",
        "#MCH 3" : "/content/MCH 3"
    }

# Catalyst weights (grams)
mch_cat_wt = {"MCH 1" : 1,
        "MCH 2" : 0.756,
        "MCH 3" : 2.54
    }

```

```

}

# TBA concentrations (mol/L)
mch_tba_conc = {"MCH 1" : 2.196,
                "MCH 2" : 4.645,
                "MCH 3" : 7.395
}

# Data indices where data is approximately linear, found by inspection
mch_linear = {"MCH 1" : 200,
              "MCH 2" : 1000,
              "MCH 3" : 0
}

# Dictionary of all initial rates
initial_rate      = {}
initial_rate_non_id = {}
error_ideal       = {}
error_slope       = {}
initial_rate["Zero"] = 0
initial_rate_non_id["Zero"] = 0
error_ideal["Zero"] = 0.0002
error_slope["Zero"] = 0.0002

# Loop over all data
for key in trial_dictionary:

    # Trial raw data
    trial_data = pd.read_csv(trial_dictionary[key])
    time       = trial_data["Time Elapsed [s]"]           # Seconds
    pressure    = trial_data["Pressure [PSI]"]            # PSI
    flow        = trial_data["Volumetric Flow [mL/min]"]  # mL/min
    temp        = trial_data["Temperature [C]"]           # Celsius

    # Linear portion of flowrate to get slope and intercept
    linear_time, linear_flow = time[linear_indices[key]:], flow[linear_indices[key]:]
    slope, intercept        = ss.linregress(linear_time, linear_flow).slope,
                             ss.linregress(linear_time, linear_flow).intercept

    # Plot results
    plt.figure(figsize = (5, 5))
    plt.title("Volumetric Flowrate of Isobutylene vs. Time (Ideal), {}".format(key), **font)
    plt.xlabel("Time [s]", **font)
    plt.ylabel("Volumetric Flowrate [mL/min]", **font)
    plt.plot(time[0:1], flow[0:1], label = "Data", color = "k")
    plt.plot(time, flow, marker = "o", markersize = 1, linestyle = "", color = "k")
    plt.plot(time, time * slope + intercept, label = "Extrapolation", linestyle = "--", color = "tab:red")
    plt.legend()
    plt.show()

    """ IDEAL GAS LAW ~~~~~
    ~~~~~"""

    # Total moles of isobutene produced, assuming Ideal Gas Law is valid
    total_volume = np.trapz(flow, time) * 1E-6 / 60          # m^3
    avg_pressure = np.mean(pressure) * 6894.76               # Pascals
    avg_temp     = np.mean(temp) + 273.15                    # Kelvin
    R            = 8.314                                     # J/mol-K
    ideal_moles  = (avg_pressure * total_volume) / (R * avg_temp) # Moles

    # Initial reaction rate (converting volumetric flowrate to molar flow rate using line intercept)
    intercept      = intercept * 1E-6 / 60                  # m^3/s
    ideal_initial_rate_per_g = (avg_pressure * intercept) / \
                               (R * avg_temp * cat_wt[key]) # mol/g-s
    ideal_initial_rate_per_mol = (avg_pressure * intercept) / \

```

```

(R * avg_temp * mol_site_per_g * cat_wt[key]) # mol/mol.cat.-s

# Error analysis for number of moles of isobutene produced and initial reaction rate (ideal)
std_pressure = 10132.5 # Pascals
std_temp = 0.01 # Kelvin
std_rate = ss.linregress(linear_time, linear_flow).intercept_stderr * \
          np.sqrt(len(linear_flow)) * 1E-6 / 6000
std_rate = (avg_pressure * std_rate) / (R * avg_temp * mol_site_per_g * cat_wt[key])

# Error in number of moles of isobutylene produced
dndp = total_volume / (R * avg_temp)
dndT = -avg_pressure * total_volume / (R * avg_temp**2)
dndV = avg_pressure / (R * avg_temp)

# Error in reaction rate
drdp = intercept / (R * avg_temp)
drdT = -avg_pressure * intercept / (R * avg_temp**2)
drdI = avg_pressure / (R * avg_temp)

error_ideal_moles = np.sqrt( (dndp)**2 * (std_pressure)**2 + (dndT)**2 * \
                             (std_temp)**2 + (dndV)**2 * (0)**2 )
error_ideal_rate = np.sqrt( (drdp)**2 * (std_pressure)**2 + (drdT)**2 * \
                             (std_temp)**2 + (drdI)**2 * (std_rate)**2 )

initial_rate[key] = ideal_initial_rate_per_mol
error_ideal[key] = error_ideal_rate
error_slope[key] = abs(slope) * 0.25 # Error due to human reaction time, ~0.25 seconds

""" NON-IDEAL (COMPRESSIBILITY FACTOR) ~~~~~
~~~~~"""

# Thermodynamic data for isobutene
Tc = 417.9 # Kelvin, https://webbook.nist.gov/cgi/cbook.cgi?ID=C115117&Mask=4
Pc = 40E5 # Pascals, https://webbook.nist.gov/cgi/cbook.cgi?ID=C115117&Mask=4
Vm = R * avg_temp / avg_pressure

# Calculate compressibility factor, z
a = 27 * R**2 * Tc**2 / (64 * Pc) # m^6 Pa mol^-2
b = R * Tc / (8 * Pc) # m^3 mol^-1
z = Vm / (Vm - b) - a / (R * avg_temp * Vm)

print("The compressibility factor is z =", z)

# Non-ideal moles produced and initial reaction rate
non_ideal_moles = (avg_pressure * total_volume) / \
                  (z * R * avg_temp) # Moles
non_ideal_initial_rate_per_g = (avg_pressure * intercept) / \
                                (z * R * avg_temp * cat_wt[key]) # mol/g-s
non_ideal_initial_rate_per_mol = (avg_pressure * intercept) / \
                                  (z * R * avg_temp * mol_site_per_g * cat_wt[key]) # mol/mol.cat.-s

initial_rate_non_id[key] = non_ideal_initial_rate_per_mol

# Error analysis for number of moles of isobutene produced and initial reaction rate (non-ideal)
error_Tc = 0.1 # Kelvin, https://webbook.nist.gov/cgi/cbook.cgi?ID=C115117&Mask=4
error_Pc = 0.1 # Pascals, https://webbook.nist.gov/cgi/cbook.cgi?ID=C115117&Mask=4

dadTc = 27 * R**2 * Tc / (32 * Pc)
dadPc = -27 * R**2 * Tc**2 / (64 * Pc**2)
dbdTc = R / (8 * Pc)
dbdPc = -R * Tc / (8 * Pc**2)

dVmdV = 1 / ideal_moles

```

```

dVmdn = -total_volume / ideal_moles**2

error_a = np.sqrt( (dadTc)**2 * (error_Tc)**2 + (dadPc)**2 * (error_Pc)**2 )
error_b = np.sqrt( (dbdTc)**2 * (error_Tc)**2 + (dbdPc)**2 * (error_Pc)**2 )
error_Vm = np.sqrt( (dVmdV)**2 * (0)**2 + (dVmdn)**2 * (error_ideal_moles)**2 )

dzdVm = -b / (Vm - b)**2 + a / (R * avg_temp * Vm**2)
dzdb = Vm / (Vm - b)**2
dzda = -1 / (R * avg_temp * Vm)
dzdT = a / (R * avg_temp**2 * Vm)
dndz = -avg_pressure * total_volume / (z**2 * R * avg_temp)

error_z = np.sqrt( (dzdVm)**2 * (error_Vm)**2 + (dzdb)**2 * (error_b)**2 + \
    (dzda)**2 * (error_a)**2 + (dzdT)**2 * (std_temp)**2 )

error_non_ideal_moles = np.sqrt( (dndp / z)**2 * (std_pressure)**2 + \
    (dndT / z)**2 * (std_temp)**2 + (dndV / z)**2 * (0)**2 + (dndz)**2 * (error_z)**2 )
error_non_ideal_rate = np.sqrt( (drdp / z)**2 * (std_pressure)**2 + \
    (drdT)**2 * (std_temp)**2 + (drdI)**2 * (std_rate)**2 + (dndz)**2 * (error_z)**2 )

# Difference between ideal and non-ideal initial reaction rates
ideal_nonideal_percent_difference = abs(ideal_initial_rate_per_g - \
    non_ideal_initial_rate_per_g) / ideal_initial_rate_per_g * 100

mch_initial_rate = []
mch_error = [0.0001]

for key in mch_data:
    # Trial raw data
    mch_trial_data = pd.read_csv(mch_data[key])
    mch_time = mch_trial_data["Time Elapsed [s]"] # Seconds
    mch_pressure = mch_trial_data["Pressure [PSI]"] # PSI
    mch_flow = mch_trial_data["Volumetric Flow [mL/min]"] # mL/min
    mch_temp = mch_trial_data["Temperature [C]"] # Celsius

    # Linear portion of flowrate to get slope and intercept
    linear_time, linear_flow = mch_time[mch_linear[key]:], mch_flow[mch_linear[key]:]
    slope, intercept, mch_rate_error = ss.linregress(linear_time, linear_flow).slope, \
        ss.linregress(linear_time, linear_flow).intercept, ss.linregress(linear_time, linear_flow).intercept_stderr

    # Plot results
    plt.figure(figsize = (5, 5))
    plt.title("Volumetric Flowrate of Isobutylene vs. Time (Ideal), {}".format(key), **font)
    plt.xlabel("Time [s]", **font)
    plt.ylabel("Volumetric Flowrate [mL/min]", **font)
    plt.plot(mch_time[0:1], mch_flow[0:1], label = "Data", color = "k")
    plt.plot(mch_time, mch_flow, marker = "o", markersize = 1, linestyle = "", color = "k")
    plt.plot(mch_time, mch_time * slope + intercept, label = "Extrapolation", linestyle = "--", color = "tab:red")
    plt.legend()
    plt.show()

    # Initial reaction rate (converting volumetric flowrate to molar flow rate using line intercept)
    intercept = intercept * 1E-6 / 60 # m^3/s
    ideal_initial_rate_per_mol = (avg_pressure * intercept) / \
        (R * avg_temp * mol_site_per_g * mch_cat_wt[key]) # mol/mol.cat.-s
    mch_rate_error = (avg_pressure * mch_rate_error) / \
        (R * avg_temp * mol_site_per_g * mch_cat_wt[key]) * 1E-6 / 60

    mch_initial_rate.append(ideal_initial_rate_per_mol)
    mch_error.append(mch_rate_error)

mch_initial_rate.append(0)

```



```

concentrations = [mch_tba_conc["MCH 1"], mch_tba_conc["MCH 2"], 0]

# Plot initial rates (Ideal)
plt.figure(figsize = (5, 5))
for key in initial_rate:
    plt.errorbar(tba_conc[key], initial_rate[key] * 100, yerr = error_ideal[key] * \
        100, color = "k", ecolor = "k", capsize = 6, marker = "s")
plt.errorbar(tba_conc[key], initial_rate[key] * 100, yerr = error_ideal[key] * \
    100, color = "k", ecolor = "k", capsize = 6, marker = "s", label = "t-BA Data")
plt.errorbar(concentrations[0:2], np.array(mch_initial_rate)[0:2] * 100,
    yerr = np.array(mch_error[:2]) * 100, marker = "o", color = "gray",
    linestyle = "", capsize = 6, label = "MCH data")

plt.xlabel("t-BA Concentration, M", **font)
plt.ylabel(r"$10^2$ \times$ IB Evo. Rate, moles/(equiv acid grps sec)", **font)
plt.title("Initial Reaction Rates", **font)
plt.yticks([0, 1, 2, 3, 4])
plt.legend()
plt.show()

# Import scikitlearn for R^2 calculation
from sklearn.metrics import r2_score

""" FINDING k' AND k ~~~~~
~~~~~"""

low_tba_conc = []
low_tba_rate = []

# Choose all trials with high water concentration
for key in h2o_conc:
    if h2o_conc[key] > 12:
        low_tba_conc.append(tba_conc[key])
        low_tba_rate.append(initial_rate[key])

# Reaction rate expression (high water concentration)
def r_prime(CA, k_prime):
    return k_prime * CA

# k' and k values
k_prime = so.curve_fit(r_prime, low_tba_conc, low_tba_rate)[0][0]
k_prime_error = np.sqrt(np.diag(so.curve_fit(r_prime, low_tba_conc, low_tba_rate)[1][0]))
k = initial_rate["Trial 1"]
k_error = error_ideal["Trial 1"]

print("k' =", k_prime, "±", k_prime_error[0][0])
print("k =", k, "±", k_error)

""" FINDING KA ~~~~~
~~~~~"""

# No water reaction rate
def no_water_rxn(CA, KA):
    return k * KA * CA / (1 + KA * CA)

KA = so.curve_fit(no_water_rxn, concentrations, mch_initial_rate,
    sigma = mch_error, absolute_sigma = True)[0][0]
KA_error = np.sqrt(np.diag(so.curve_fit(no_water_rxn, concentrations,
    mch_initial_rate, sigma = mch_error, absolute_sigma = True)[1][0]))

print("K_A =", KA, "±", KA_error[0][0])

""" FINDING KW AND PLOTTING ~~~~~
~~~~~"""

# Initialize K_W and R^2 lists
KW_all = []

```

```

KW_error    = []
R_sq        = []
whole_range = np.linspace(0, 10, 1000)

# Loop over all gamma values to find which one has highest R^2 value
for gamma in range(1, 4):

    # Reaction rate expression (medium TBA concentration)
    def r(CA, KW):
        CW = 55.5 - CA * 55.5 / 10

        return k * KA * CA / (1 + KA * CA + (KW * CW)**gamma) + k_prime * CA

    KW = so.curve_fit(r, list(tba_conc.values()), list(initial_rate.values()),
        sigma = list(error_ideal.values()), absolute_sigma = True, bounds = (0, 5))[0][0]
    KW_std = np.sqrt(np.diag(so.curve_fit(r, list(tba_conc.values()), list(initial_rate.values()),
        sigma = list(error_ideal.values()), absolute_sigma = True, bounds = (0, 5))[1][0]))
    KW_all.append(KW)
    KW_error.append(KW_std[0][0])

# Had to force KW to be within 0 and 5 because if I didn't, there would be a KW of over 100
print(KW_all)
print(KW_error)

# Full reaction rate equation
def r_tot(CA, KA, KW, n):
    CW = 55.5 - CA * 55.5 / 10

    return (k * KA * CA / (1 + KA * CA + (KW * CW)**n) + k_prime * CA) * 100

# Plot all data
for gamma in range(3):
    # Plot initial rates (Ideal)
    plt.figure(figsize = (3.5, 3.5))
    for key in initial_rate:
        plt.errorbar(tba_conc[key], initial_rate[key] * 100, yerr = error_ideal[key] * \
            100, color = "k", ecolor = "k", capsize = 6, marker = "s")
    plt.errorbar(tba_conc[key], initial_rate[key] * 100, yerr = error_ideal[key] * \
        100, color = "k", ecolor = "k", capsize = 6, marker = "s", label = "t-BA Data")
    plt.errorbar(concentrations[0:2], np.array(mch_initial_rate)[0:2] * 100,
        yerr = np.array(mch_error[:2]) * 100, marker = "o", color = "gray",
        linestyle = "", capsize = 6, label = "MCH data")
    plt.plot(whole_range, r_tot(whole_range, KA, KW_all[gamma], gamma + 1),
        linestyle = "--", color = "k")
    plt.plot(whole_range, no_water_rxn(whole_range, KA) * 100, linestyle = "--", color = "gray")
    plt.xlabel("t-BA Concentration, M", **font)
    plt.ylabel(r"$10^{-2}$ \times IB Evo. Rate, moles/(equiv acid grps sec)", **font)
    plt.title("Initial Reaction Rates, gamma = {}".format(gamma + 1), **font)
    plt.yticks([0, 1, 2, 3, 4])
    plt.legend()
    plt.show()

# Plot only beginning linear portion
for gamma in range(3):
    # Plot initial rates (Ideal)
    plt.figure(figsize = (3.5, 3.5))
    for key in initial_rate:
        plt.errorbar(tba_conc[key], initial_rate[key] * 100,
            yerr = error_ideal[key] * 100, color = "k", ecolor = "k", capsize = 6, marker = "s")
    plt.plot(whole_range, r_tot(whole_range, KA, KW_all[gamma], gamma + 1), linestyle = "--", color = "k")
    plt.xlabel("t-BA Concentration, M", **font)
    plt.ylabel(r"$10^{-2}$ \times IB Evo. Rate, moles/(equiv acid grps sec)", **font)
    plt.title("Linear Portion of Initial Reaction Rates, gamma = {}".format(gamma + 1), **font)
    plt.xlim(xmin = 0, xmax = 7.32)
    plt.ylim(ymin = 0, ymax = 0.5)
    plt.show()

```

```

# Calculate sum of squares error (or residual sum of squares)
rss1 = 0
rss2 = 0
rss3 = 0

# Find Sum of Squares error
for key in initial_rate:
    rss1 += (initial_rate[key] - r_tot(tba_conc[key], KA, KW_all[0], 1)/100)**2
    rss2 += (initial_rate[key] - r_tot(tba_conc[key], KA, KW_all[1], 2)/100)**2
    rss3 += (initial_rate[key] - r_tot(tba_conc[key], KA, KW_all[2], 3)/100)**2

print("RSS for gamma = 1:", rss1)
print("RSS for gamma = 2:", rss2)
print("RSS for gamma = 3:", rss3)
print()

""" FINDING OPTIMUM GAMMA ~~~~~
~~~~~"""

# Rate equation to fit only gamma
def find_gamma(CA, gamma):
    CW = 55.5 - CA * 55.5 / 10

    return (k * KA * CA / (1 + KA * CA + (KW_all[1] * CW)**gamma) + k_prime * CA)

optimal_gamma = np.round(so.curve_fit(find_gamma, list(tba_conc.values()),
list(initial_rate.values()), sigma = list(error_ideal.values()), absolute_sigma = True)[0][0], 2)
print("The 'optimal' value for gamma:", optimal_gamma)

# Calculate sum of squares error (or residual sum of squares)
rss_gamma = 0

# Find Sum of Squares error
for key in initial_rate:
    rss_gamma += (initial_rate[key] - r_tot(tba_conc[key], KA, KW_all[1], optimal_gamma)/100)**2

print("RSS for gamma = {}".format(optimal_gamma), rss_gamma)

plt.figure(figsize = (5, 5))
for key in initial_rate:
    plt.errorbar(tba_conc[key], initial_rate[key] * 100, yerr = error_ideal[key] * 100,
        color = "k", ecolor = "k", capsize = 6, marker = "s")

plt.errorbar(concentrations[0:2], np.array(mch_initial_rate)[0:2] * 100,
    yerr = np.array(mch_error[:2]) * 100, marker = "o", color = "gray", linestyle = "", capsize = 6)
plt.plot(whole_range, find_gamma(whole_range, optimal_gamma) * 100, linestyle = "--", color = "k")
plt.plot(whole_range, no_water_rxn(whole_range, KA) * 100, linestyle = "--", color = "gray")
plt.xlabel("t-BA Concentration, M", **font)
plt.ylabel(r"$10^{-2}$ IB Evo. Rate, moles/(equiv acid grps sec)", **font)
plt.title("Initial Reaction Rates, gamma = {}".format(optimal_gamma), **font)
plt.show()

# Plot only beginning linear portion
plt.figure(figsize = (5, 5))
for key in initial_rate:
    plt.errorbar(tba_conc[key], initial_rate[key] * 100, yerr = error_ideal[key] * 100,
        color = "k", ecolor = "k", capsize = 6, marker = "s")
plt.plot(whole_range, r_tot(whole_range, KA, KW, optimal_gamma), linestyle = "--", color = "k")
plt.xlabel("t-BA Concentration, M", **font)
plt.ylabel(r"$10^{-2}$ IB Evo. Rate, moles/(equiv acid grps sec)", **font)
plt.title("Linear Portion of Initial Reaction Rates, gamma = {}".format(optimal_gamma), **font)
plt.xlim(xmin = 0, xmax = 7.32)
plt.ylim(ymin = 0, ymax = 0.5)
plt.show()

```