Overpressure Analysis of Girard Point Refinery Accident

Boiling-Liquid Expanding-Vapor Explosion

Author:
S. Kevin McNeill
shonn.mcneill@atf.gov

National Center for Explosives Training and Research
Explosives Research and Development Division

July 1, 2019

Contents

List of Figures

List of Tables

List of Codes

1 Introduction

On June 21, 2019, an accidental fire and explosion occurred at the Girard Point Refinery owned by Philadelphia Energy Solutionsa. One of the three explosions observed was a storage tank weighing 74,000 lb containing 90,000 lb of iso-butane, propane, and hydrofluoric acid. The tank was propelled approximately 2,000 ft from the blast seat. The Philadelphia Fire Department requested ATF estimate the blast overpressure generated when the tank exploded.

It is hypothesized that a boiling-liquid expanding-vapor explosion (BLEVE) event provided the energy to generate a blast wave. This notebook is an engineering analysis to estimate the blast overpressure. The analysis is based upon an adiabatic but irreversible energy analysis developed by Planas-Cuchi et.al [1].

2 Background

This analysis calculates the pressure of the blastwave generated from the BLEVE by finding the equivalent mass of TNT that would release the same amount of energy. With the mass of TNT known, a pressure can be calculated using the Kingery-Bulmash equations. The energy released from the BLEVE is calculated using the 1st Law of Thermodynamics assuming an adiabatic irreversible process.

Boiling-Liquid Expanding-Vapor Explosion A BLEVE results from the sudden failure of a tank containing a compressed vapor (head space) and a super-heated liquid (a liquid heated above it's boiling point but without boiling). The magnitude of the blast depends on how super-heated the liquid was at failure. As the level of super-heat rises, the portion of liquid that flash-boils rises, thus increasing the energy released. Once containment failure occurs the energy is distributed into four forms:

1. Overpressure wave
2. Kinetic energy of fragments
3. Deformation and failure of the containment material
4. Heat transferred to environment

The distribution of the energy into the these four forms depends on the specifics of the explosion. Casal et al. found that a *fragile* failure releases 80% of the energy into the blastwave, while a *ductile* failure releases 40% of the energy into the blastwave. The remaining energy becomes kinetic energy of the fragments. The heat transfer to the environment is relatively small.[2] In practice most pressure vessels are designed with materials that are ductile rather than brittle to avoid sudden and catastrophic brittle (fragile) failures [3].

2.1 First Law of Thermodynamics

The 1st Law of Thermodynamics is a restatement of the conservation of energy (energy cannot be created or destroyed). A way of stating the first law of thermodynamics is the internal energy (ΔU) of a system is given by the sum of the heat (Q) that flows into and out-of the system and the work (W) done on the system:

$$\Delta U = Q + W \tag{2.1}$$

Therefore, there are only two ways to change the internal energy ΔU of the system:

- Add or remove heat from the system, Q.
- Perform or extract work from the system, W.

It is assumed for this analysis the BLEVE expansion of the super-heated liquid (initial state) to atmospheric conditions (final state) occurs so quickly there is no time for heat to be added or removed from the system. When heat cannot be added or removed from a system it is called an adiabatic process. During an adiabatic process the heat (Q) is zero and the 1st Law of Thermodynamics can be rewritten to:

$$\Delta U = W \tag{2.2}$$

Work The work W of an expanding gas is given by,

$$W = P_0 \Delta V \tag{2.3}$$

Where P_0 is the initial pressure and ΔV is the difference in the final and initial volumes. In the case of the BLEVE the change in volume is due to the conversion of the super-heated liquid to a gas. We know the initial volume of the system, V_i in this case, the tank volume. However, the final volume, V_f, is an unknown. We can however transform the final volume into terms we do know. If we rewrite the final

volume (V_f) in terms of the specific volume (volume divided by mass) times the mass we can transform our change in final volume into a change in final mass,

$$\Delta V = V_f - V_i \tag{2.4}$$

$$\Delta V = (\nu_G m_G + \nu_L m_L)_f - V_i \tag{2.5}$$

Therefore,

$$W = P_0 \left[(\nu_G m_G + \nu_L m_L)_f - V_i \right] \tag{2.6}$$

The quality of a gas-liquid mixture is defined as the fraction of the total mass (gas + liquid) that is saturated vapor,

$$\chi = \frac{m_G}{m_T} \tag{2.7}$$

Where m_G is the mass of the gas and m_T is the total mass. Solving for the mass of the gas,

$$m_G = m_T \chi \tag{2.8}$$

The mass of the liquid is then given by,

$$m_T = m_G + m_L \tag{2.9}$$

$$m_L = m_T - m_G \tag{2.10}$$

$$m_L = m_T - m_T \chi \tag{2.11}$$

Substituting equations (7) and (10) back into (5) we have,

$$W = P_0 \left[(\nu_G m_T \chi + \nu_L (m_T - m_T \chi))_f - V_i \right] \tag{2.12}$$

$$W = P_0 \left[(\nu_G m_T \chi + \nu_L m_T - \nu_L m_T \chi)_f - V_i \right] \tag{2.13}$$

$$W = P_0 \left[((\nu_G - \nu_L) m_T \chi + \nu_L m_T)_f - V_i \right] \tag{2.14}$$

2.2 Internal Energy

The internal energy is the energy of atomic motion. It is the sum of the molecular kinetic and potential energies in a system (in our case, the tank filled with fluid). The molecular kinetic energy is made up of three types of molecular motion:

- translational,
- rotational, and
- vibrational.

The molecular potential energy is derived from the electromagnetic force that exists between the atoms in individual molecules and the intermolecular forces between molecules. The internal energy is directly proportional to the temperature of the system (higher temperatures mean more internal energy and lower temperatures mean lower internal energy).

The change in internal energy can be changed similarly to how we change the volume in the Work section. The change in internal energy can be rewritten in terms of the specific internal energy (internal energy divided by the mass) and the quality,

$$\Delta U = U_f - U_i \tag{2.15}$$

$$\Delta U = (u_G m_G + u_L m_L)_f - U_i \tag{2.16}$$

Substituting equations (7) and (10) into the internal energy equation (15) we have,

$$\Delta U = (u_G m_T \chi + u_L (m_T - m_T \chi))_f - U_i \tag{2.17}$$

$$\Delta U = (u_G m_T \chi + u_L m_T - u_L m_T \chi)_f - U_i \tag{2.18}$$

$$\Delta U = ((u_G - u_L) m_T \chi + u_L m_T)_f - U_i \tag{2.19}$$

Referring back to equation (3) and substituting equations (14) and (19) we can solve for the quality χ,

$$\chi = \frac{m_T P_0 v_L - V_i P_0 + m_T u_L - U_i}{[(u_L - u_G) - (v_G - v_L)P_0] m_T} \tag{2.20}$$

By substituting χ back into equation (19) we can solve for the internal energy, ΔU. Once we know the change in internal energy due to the flash-boiling we can calculate the equivalent weight of TNT that will produce the same change in internal energy.

2.3 Conversion to TNT Equivalent Weight

The equation for the conversion to equivalent TNT weight in kilograms is,

$$W_{TNT} = \beta \, 0.2136 \Delta U \tag{2.21}$$

Where: - β is the fraction of the energy released converted to the blast wave (fragile = 80% and ductile = 40%) - $0.2136 \frac{kg}{MJ}$ is the inverse of blast energy of TNT $(4680 \, J/g_{TNT})$[4] - ΔU is the internal energy of the flash boiling liquid in $bar \, m^3$.

With the weight of TNT known it is a simple matter to calculate the overpressure at a specific distance from the BLEVE using the Kingery Bulmash equations [5].

3 Notebook Imports

```python
# Data manipulation
import pandas as pd
import numpy as np

# Options for pandas
pd.options.display.max_columns = 50
pd.options.display.max_rows = 30

# Display all cell outputs
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = 'all'

from IPython import get_ipython
ipython = get_ipython()

# autoreload extension
if 'autoreload' not in ipython.extension_manager.loaded:
    %load_ext autoreload

%autoreload 2

# Visualizations
import plotly as py
import plotly.graph_objs as go
import plotly.figure_factory as ff
from plotly.offline import iplot, init_notebook_mode
init_notebook_mode(connected=True)

import cufflinks as cf
cf.go_offline(connected=True)
cf.set_config_file(theme='white')

# Units and Thermodynamic Data
import pint as unit
import pyromat as pm
```

```python
# Define a dictionary containing the study data
verify_data = {'parameter': [
                'Pressure [kPa]', 'Temperature [C]',
                'Total mass [kg]', 'Mass of liquid [kg]',
                'Mass of vapour [kg]', 'Vapour specific volume [
    ↪ m3 kg-1]',
                'Liquid specific volume [m3 kg-1]','System
    ↪ specific volume[m3 kg-1]',
                'Total vapour volume [m3]', 'Total liquid volume
    ↪  [m3]',
                'Total volume [m3]', 'Vapour fraction',
                'Vapour specific internal energy [kJ kg-1]', '
    ↪ Liquid specific internal energy [kJ kg-1]',
                'System specific internal energy [kJ kg-1]', '
```

```
↪ Total vapour internal energy [MJ]',
                    'Total liquid internal energy [MJ]', 'Total
↪ internal energy [MJ]',
                    'Vapour specific entropy [kJ kg-1 K-1]', 'Liquid
↪ specific entropy [kJ kg-1 K-1]',
                    'System specific entropy [kJ kg-1 K-1]'],
                    'explosion_state': [1901, 55, 100956, 100007,
↪ 949.1,0.02293, 0.002282, 0.002476, 21.8, 228.2, 250,
↪ 0.009401, 582, 349.2, 351.4, 560, 34920, 35480, 2.33, 1.501,
↪ 1.508],
                    'final_state': [101.3, -42.02, 100956, 41288,
↪ 59668, 0.4136, 0.001721, 0.2452, 24681, 71, 24752, 0.591,
↪ 483.7, 100.1, 326.8, 28860, 4133, 32990, 2.448, 0.6068,
↪ 1.695]}
# Convert the dictionary into DataFrame
df = pd.DataFrame(verify_data)
df.set_index('parameter');
```

```
def delta_U(uL, uG, mT, x, Ui):
    '''
    This function calculates the change in internal energy
    of an adiabatically expanding liquid.  Where u is the
    specific internal energy, L is liquid, G is gas,
    mT is total mass, x is thermodynamic quality, and Ui
    is the internal energy at the initial state.
    '''
    return (uL-uG)*mT*x-mT*uL+Ui

def P_delta_V(P0, vL, vG, mT, x, Vi):
    '''
    This function calculates the P(Vf-Vi)
    of an adiabatically expanding liquid.  Where v is the
    specific volume, L is liquid, G is gas,
    mT is total mass, x is thermodynamic quality,
    P0 is the atmospheric pressure,and Vi
    is the volume at the initial state.
    '''
    return P0*((vG-vL)*mT*x+mT*vL-Vi)

def X(uL, uG, mT, x, Ui, P0, vL, vG, Vi):
    '''
    This function calculates the quality
    of an adiabatically expanding liquid.  Where u is the
    specific internal energy, v is the specific volume,
    L is liquid, G is gas, mT is total mass,
    x is thermodynamic quality, P0 is the atmospheric and
    Ui is the internal energy at the initial state.
    '''
    return (mT*P0*vL-Vi*P0+mT*uL-Ui)/((uL-uG)-(vG-vL)*P0)*mT
```

```
df.loc['Pressure [kPa]']['explosion_state']
```

```
  --------------------------------------------------------------------------
```

```
 TypeError                                          Traceback (most recent
↪ call last)
```

```
 <ipython-input-9-5db034eab431> in <module>
 ----> 1 df.iloc['Pressure [kPa]']['explosion_state']
```

```
 ~/miniconda3/envs/philadelphia_refinery/lib/python3.7/site-packages
↪ /pandas/core/indexing.py in __getitem__(self, key)
    1498
    1499               maybe_callable = com.apply_if_callable(key,
↪ self.obj)
 -> 1500               return self._getitem_axis(maybe_callable, axis=
↪ axis)
    1501
    1502       def _is_scalar_access(self, key):
```

```
 ~/miniconda3/envs/philadelphia_refinery/lib/python3.7/site-packages
↪ /pandas/core/indexing.py in _getitem_axis(self, key, axis)
    2224            else:
    2225                if not is_integer(key):
 -> 2226                    raise TypeError("Cannot index by location
↪ index with a "
    2227                                    "non-integer key")
    2228
```

```
  TypeError: Cannot index by location index with a non-integer key
```

```python
propane = pm.get('ig.C3H8')

T = list(range(300, 4000))
data = [go.Scatter(x=T,y=propane.e(T))]
layout = go.Layout(xaxis={'title':'Temperature (K)'},
                   yaxis={'title':'Internal Energy, U (kJ/kg)'})
figure=go.Figure(data=data,layout=layout)
py.offline.iplot(figure)
```

4 Cleanup Data

Like families, tidy datasets are all alike but every messy dataset is messy in its own way. Tidy datasets provide a standardized way to link the structure of a dataset (its physical layout) with its semantics (its meaning). In this section, I'll provide some standard vocabulary for describing the structure and semantics of a dataset, and then use those definitions to define tidy data. tidyr

5 Analysis/Modeling

6 Results

- Focus on main analysis steps and findings
- Remove intermediate results or move to supplement

7 Conclusion

8 References

1. Planas-Cuchi, E., Salla, J. M., & Casal, J. (2004). Calculating overpressure from BLEVE explosions. Journal of Loss Prevention in the Process Industries, 17(6), 431–436. https://doi.org/10.1016/j.jlp.2004.08.002